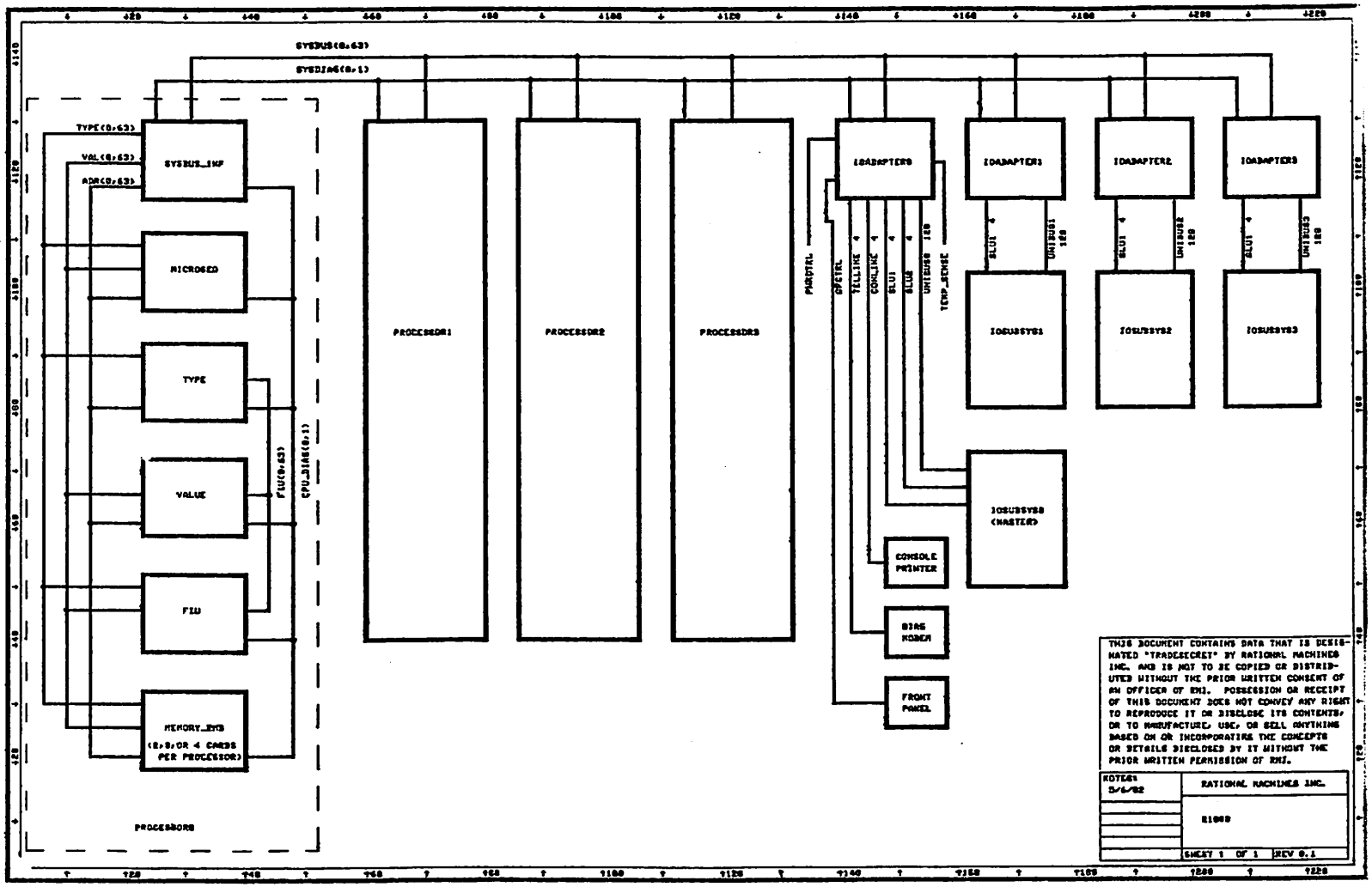
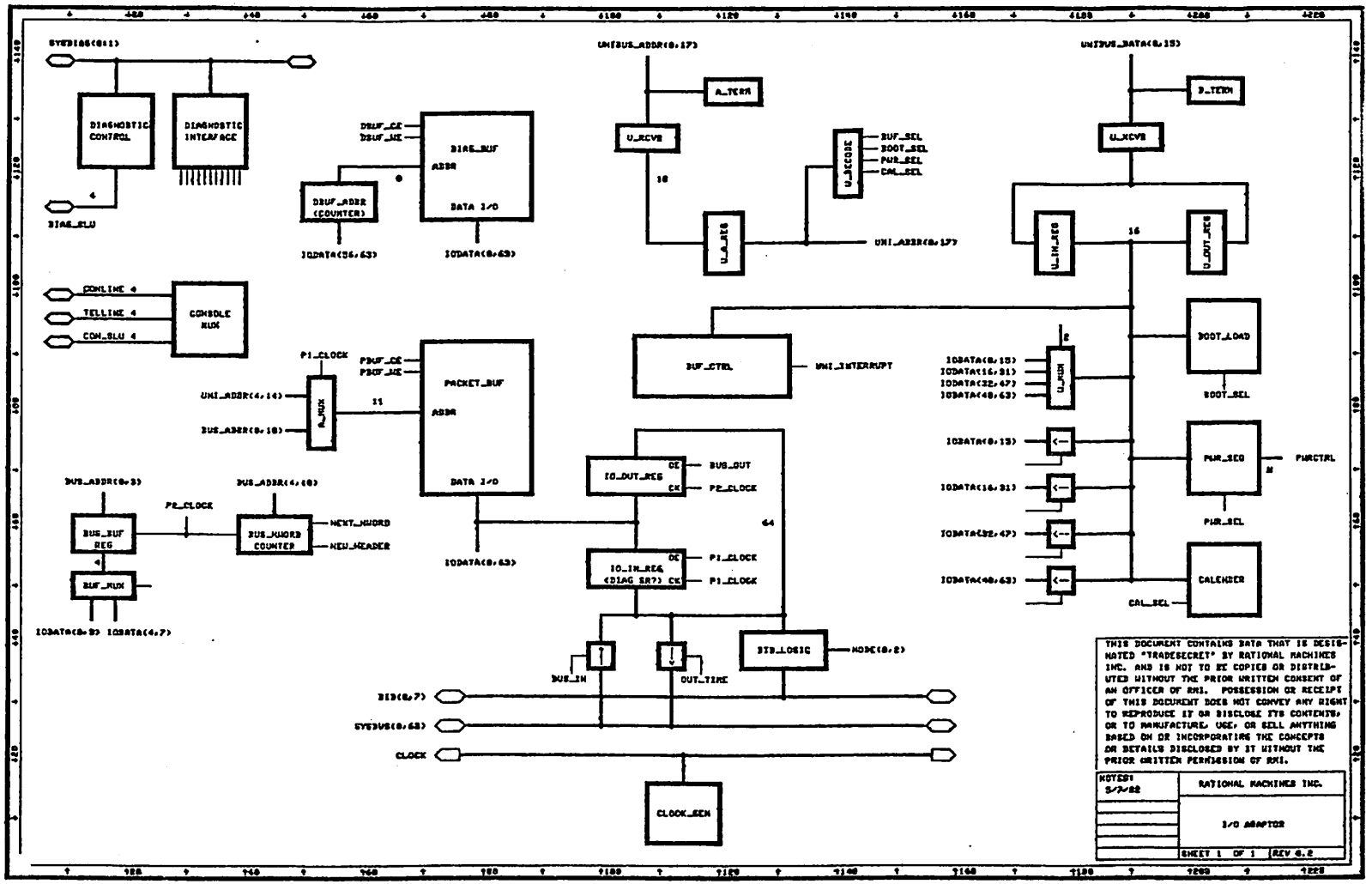
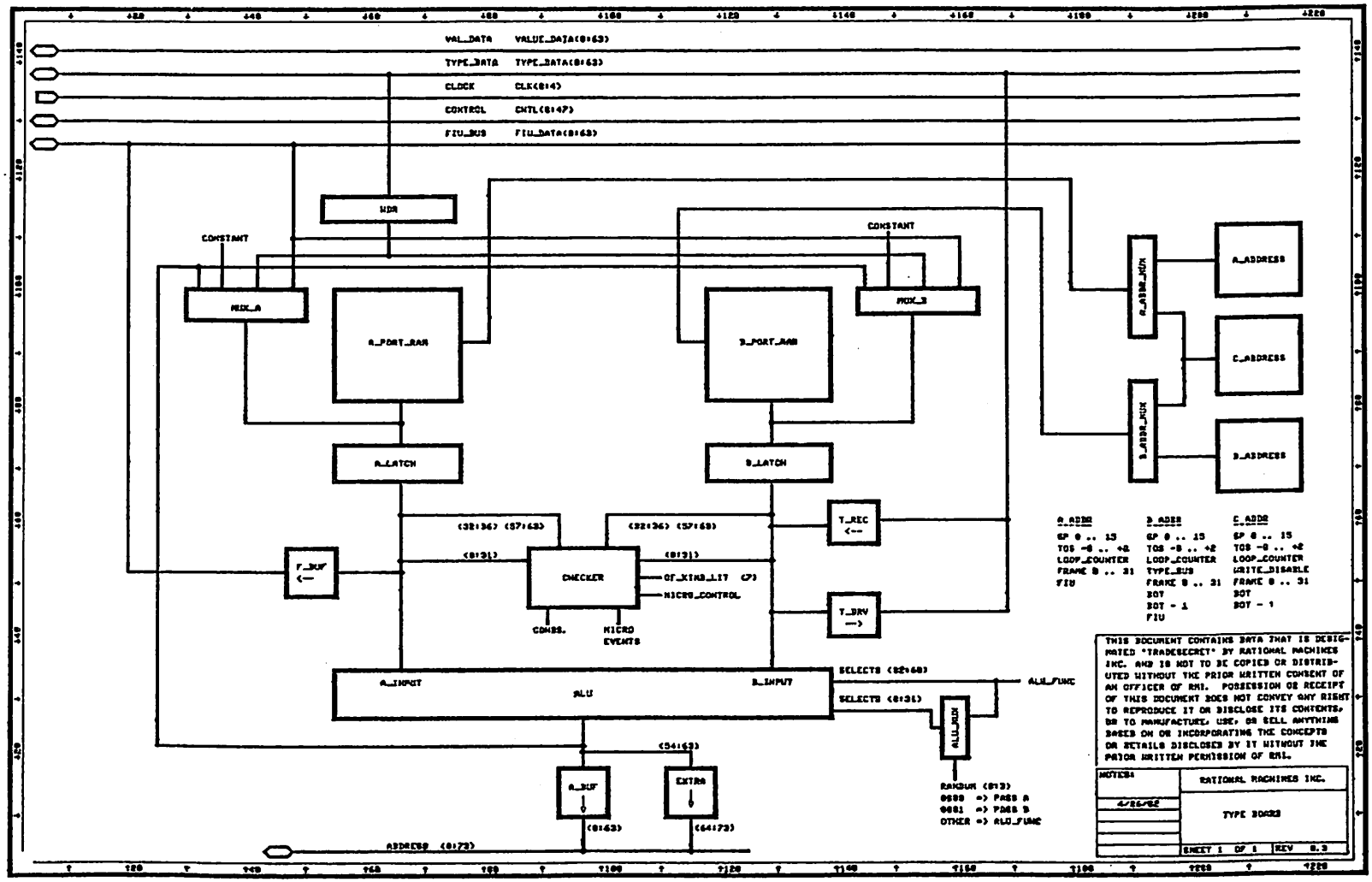
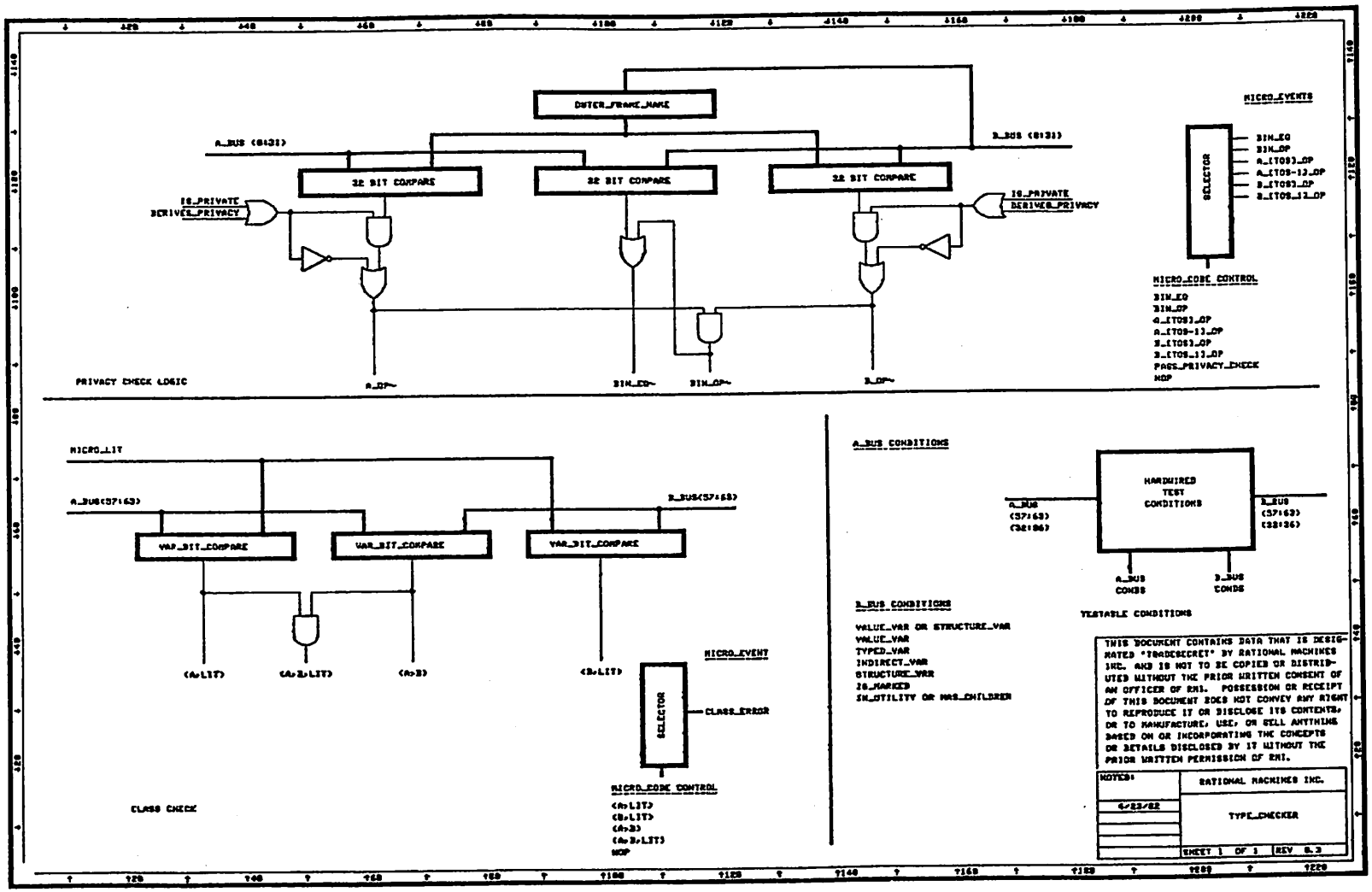
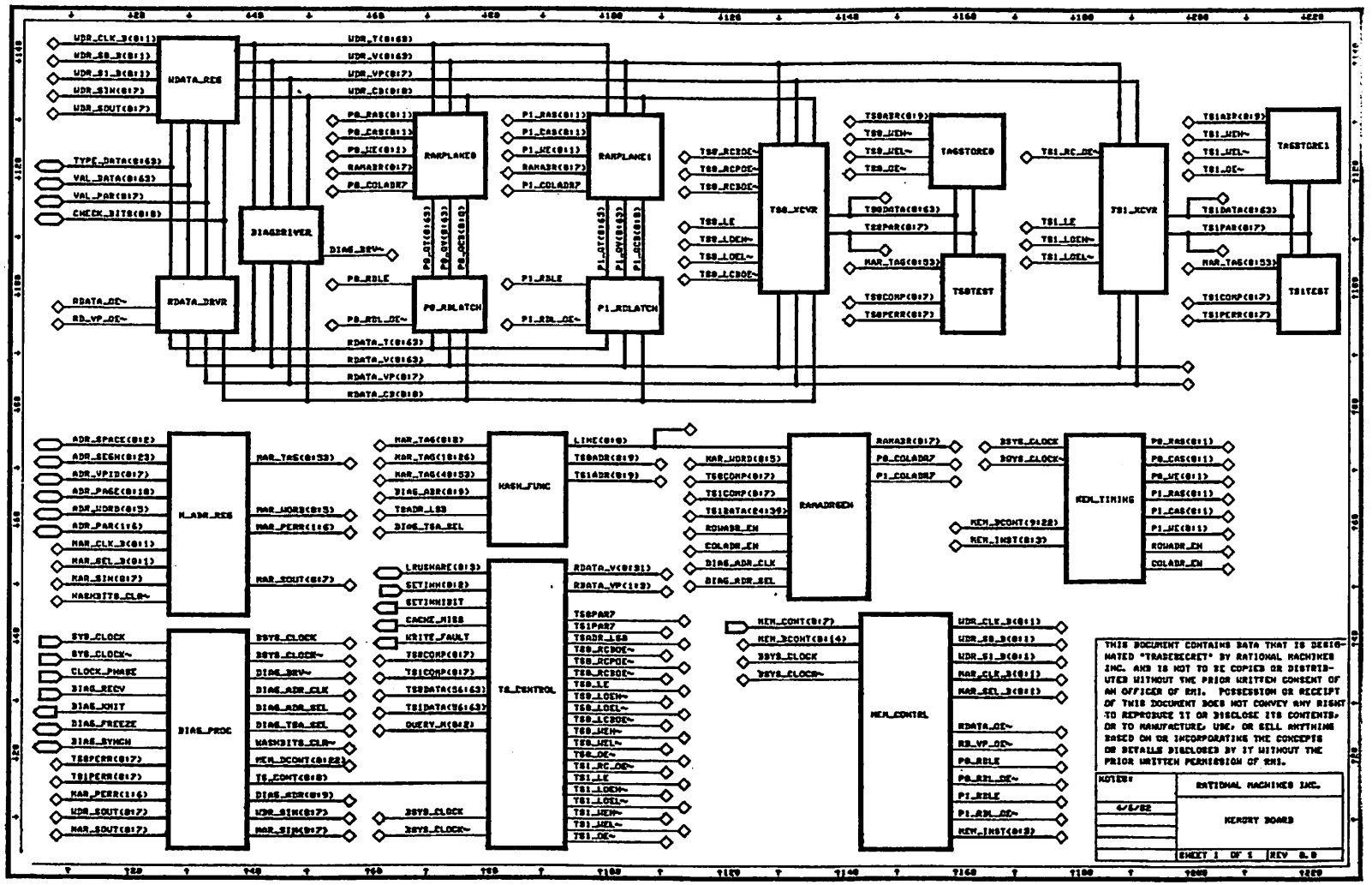
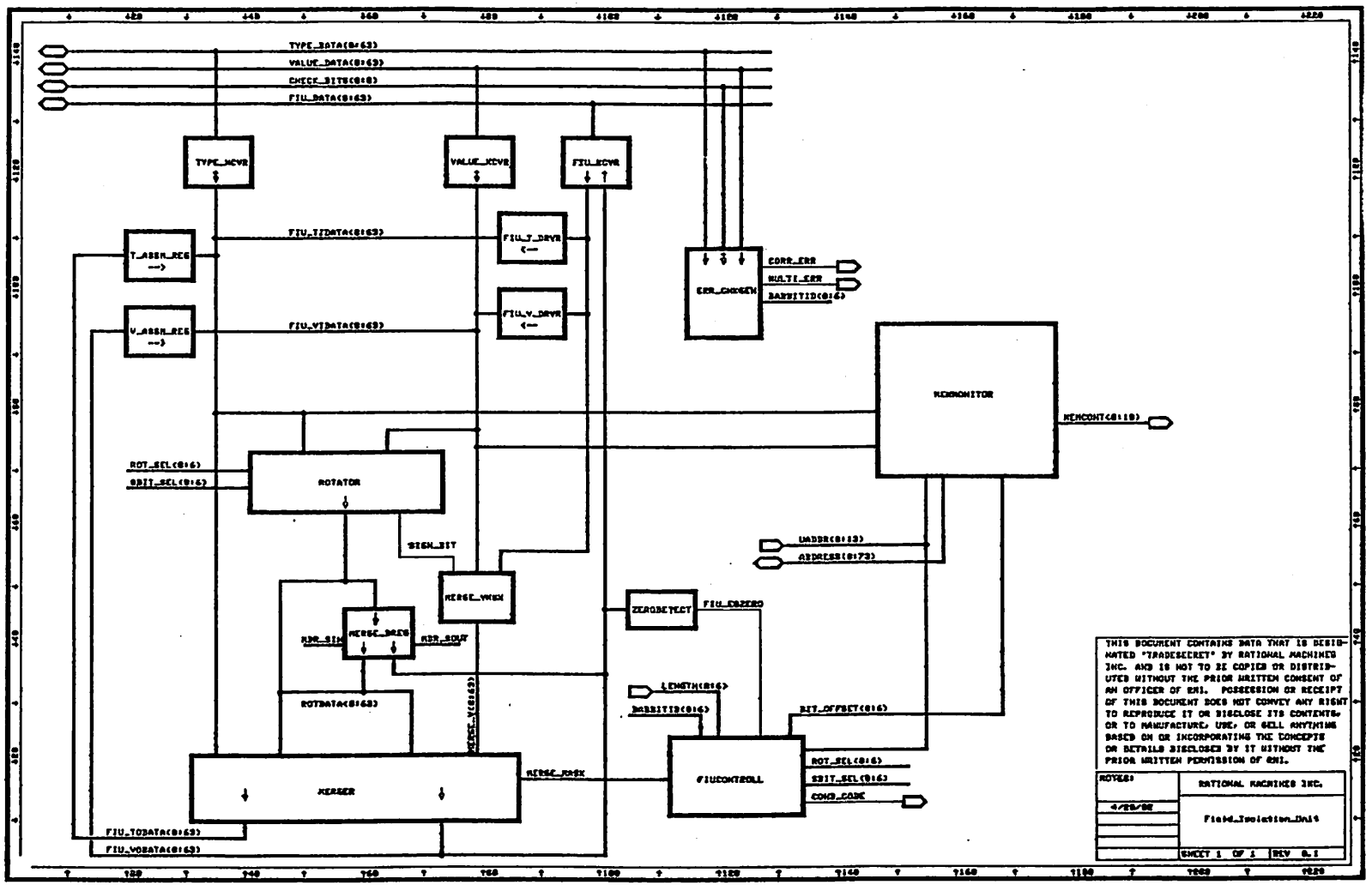

SYSBUS(0:63)
SYSDIAG(0:1)
TYPE(0:63)
VAL(0:63)
ADR(0:63)
SYSBUS_INF
MICROSEQ
TYPE
VALUE
FIU
MEMORY_SYS
(2,3,OR 4 CARDS PER PROCESSOR)
FIU(0:63)
CPU_DIAG(0:1)
PROCESSOR
PROCESSOR1
PROCESSOR2
PROCESSOR3
IOADAPTER0
IOADAPTER1
IOADAPTER2
IOADAPTER3
PORTAL
SPETEL
TELLINE
CONLINE
BLUI
BLUB
UNIBUS0
TEMP_SENSE
BLUI
UNIBUS1
IOSUBSYS1
IOSUBSYS2
IOSUBSYS3
IOSUBSYS0 (MASTER)
CONSOLE PRINTER
BIAS MODEM
FRONT PANEL
THIS DOCUMENT CONTAINS DATA THAT IS DESIGNATED "TRADESECRET" BY RATIONAL MACHINES INC. AND IS NOT TO BE COPIED OR DISTRIBUTED WITHOUT THE PRIOR WRITTEN CONSENT OF AN OFFICER OF RMI. POSSESSION OR RECEIPT OF THIS DOCUMENT DOES NOT CONVEY ANY RIGHT TO REPRODUCE IT OR DISCLOSE ITS CONTENTS, OR TO MANUFACTURE, USE, OR SELL ANYTHING BASED ON OR INCORPORATING THE CONCEPTS OR DETAILS DISCLOSED BY IT WITHOUT THE PRIOR WRITTEN PERMISSION OF RMI.
NOTES:
5/6/82
RATIONAL MACHINES INC.
81069
SHEET 1 OF 1
REV 0.1


SYEDIAG(0:1)
DIAGNOSTIC CONTROL
DIAGNOSTIC INTERFACE
4
DIAG_SLU
DBUF_OE
DBUF_WE
DIAG_BUF
ADDR
DATA I/O
DBUF_ADDR (COUNTER)
IODATA(56:63)
IODATA(0:63)
CONLINE 4
TELLINE 4
CON_SLU 4
CONSOLE MUX
P1_CLOCK
PBUF_OE
PBUF_WE
PACKET_BUF
ADDR
DATA I/O
UNI_ADDR(4:14)
BUS_ADDR(0:10)
A_MUX
11
IODATA(0:63)
BUS_ADDR(0:3)
BUS_ADDR(4:10)
PB_CLOCK
BUS_BUF REG
BUS_WORD COUNTER
NEXT_WORD
NEW_HEADER
4
BUF_RUN
IODATA(0:3) IODATA(4:7)
UNIBUS_ADDR(0:17)
A_TERM
U_RCVR
18
U_ACCODE
BUF_SEL
BOOT_SEL
PWR_SEL
CAL_SEL
U_A_REG
UNI_ADDR(0:17)
UNIBUS_DATA(0:15)
B_TERM
U_XCVR
U_IN_REG
16
U_OUT_REG
BUF_CTRL
UNI_INTERRUPT
2
U_RUN
IODATA(0:15)
IODATA(16:31)
IODATA(32:47)
IODATA(48:63)
BOOT_LOAD
BOOT_SEL
PWR_REG
PWRCTRL
M
PWR_SEL
CALENDER
CAL_SEL
IO_OUT_REG
OE
CK
BUS_OUT
PB_CLOCK
64
IO_IN_REG (DIAG SR?)
P1_CLOCK
BUS_IN
OUT_TIME
STB_LOGIC
MODE(0:2)
BID(0:7)
SYEBUS(0:63)
CLOCK
CLOCK_GEN
THIS DOCUMENT CONTAINS DATA THAT IS DESIGNATED "TRADESECRET" BY RATIONAL MACHINES INC. AND IS NOT TO BE COPIED OR DISTRIBUTED WITHOUT THE PRIOR WRITTEN CONSENT OF AN OFFICER OF RMI. POSSESSION OR RECEIPT OF THIS DOCUMENT DOES NOT CONVEY ANY RIGHT TO REPRODUCE IT OR DISCLOSE ITS CONTENTS, OR TO MANUFACTURE, USE, OR SELL ANYTHING BASED ON OR INCORPORATING THE CONCEPTS OR DETAILS DISCLOSED BY IT WITHOUT THE PRIOR WRITTEN PERMISSION OF RMI.
NOTES!
5/7/82
RATIONAL MACHINES INC.
I/O ADAPTOR
SHEET 1 OF 1
REV 0.2


VAL_DATA VALUE_DATA(0:63)
TYPE_DATA TYPE_DATA(0:63)
CLOCK CLK(0:4)
CONTROL CNTL(0:4P)
FIU_BUS FIU_DATA(0:63)
MDR
CONSTANT
CONSTANT
MUX_A
MUX_B
A_PORT_RAM
B_PORT_RAM
A_LATCH
B_LATCH
A_ADDR_MUX
B_ADDR_MUX
A_ADDRESS
C_ADDRESS
B_ADDRESS
(32:36) (57:63)
(32:36) (57:63)
(0:31)
(0:31)
T_REC
<--
T_DRV
-->
F_BUF
<--
CHECKER
OF_XTND_LIT (7)
MICRO_CONTROL
COMBS.
MICRO EVENTS
A_INPUT
ALU
B_INPUT
SELECTS (32:63)
SELECTS (0:31)
ALU_FUNC
ALU_MUX
(54:63)
A_BUF
EXTRA
(0:63)
(64:73)
ADDRESS (0:73)
RANDOM (0:3)
0000 => PASS A
0001 => PASS B
OTHER => ALU_FUNC
A ADDR
GP 0 .. 15
TOS -8 .. +8
LOOP_COUNTER
FRAME 0 .. 31
FIU
B ADDR
GP 0 .. 15
TOS -8 .. +8
LOOP_COUNTER
TYPE_BUS
FRAME 0 .. 31
BOT
BOT - 1
FIU
C ADDR
GP 0 .. 15
TOS -8 .. +8
LOOP_COUNTER
WRITE_DISABLE
FRAME 0 .. 31
BOT
BOT - 1
THIS DOCUMENT CONTAINS DATA THAT IS DESIGNATED "TRADESECRET" BY RATIONAL MACHINES INC. AND IS NOT TO BE COPIED OR DISTRIBUTED WITHOUT THE PRIOR WRITTEN CONSENT OF AN OFFICER OF RMI. POSSESSION OR RECEIPT OF THIS DOCUMENT DOES NOT CONVEY ANY RIGHT TO REPRODUCE IT OR DISCLOSE ITS CONTENTS, OR TO MANUFACTURE, USE, OR SELL ANYTHING BASED ON OR INCORPORATING THE CONCEPTS OR DETAILS DISCLOSED BY IT WITHOUT THE PRIOR WRITTEN PERMISSION OF RMI.
NOTES:
RATIONAL MACHINES INC.
4/26/82
TYPE BOARD
SHEET 1 OF 1
REV 0.3


OUTER_FRAME_NAME
A_BUS (64:31)
B_BUS (64:31)
32 BIT COMPARE
32 BIT COMPARE
32 BIT COMPARE
IS_PRIVATE
DERIVES_PRIVACY
IS_PRIVATE
DERIVES_PRIVACY
A_OP
BIN_EQ
BIN_OP
B_OP
PRIVACY CHECK LOGIC
MICRO_EVENTS
SELECTOR
BIN_EQ
BIN_OP
A_(TOS)_OP
A_(TOS-1)_OP
B_(TOS)_OP
B_(TOS-1)_OP
MICRO_CODE CONTROL
BIN_EQ
BIN_OP
A_(TOS)_OP
A_(TOS-1)_OP
B_(TOS)_OP
B_(TOS-1)_OP
PASS_PRIVACY_CHECK
NOP
MICRO_LIT
A_BUS(57:63)
B_BUS(57:63)
VAR_BIT_COMPARE
VAR_BIT_COMPARE
VAR_BIT_COMPARE
(A<LIT)
(A>B<LIT)
(A>B)
(B<LIT)
MICRO_EVENT
SELECTOR
CLASS_ERROR
MICRO_CODE CONTROL
(A<LIT)
(B<LIT)
(A>B)
(A>B<LIT)
NOP
CLASS CHECK
A_BUS CONDITIONS
HARDWIRED TEST CONDITIONS
A_BUS (57:63) (32:36)
B_BUS (57:63) (32:36)
A_BUS CONDS
B_BUS CONDS
TESTABLE CONDITIONS
B_BUS CONDITIONS
VALUE_VAR OR STRUCTURE_VAR
VALUE_VAR
TYPED_VAR
INDIRECT_VAR
STRUCTURE_VAR
IS_MARKED
IS_UTILITY OR HAS_CHILDREN
THIS DOCUMENT CONTAINS DATA THAT IS DESIGNATED "TRADESECRET" BY RATIONAL MACHINES INC. AND IS NOT TO BE COPIED OR DISTRIBUTED WITHOUT THE PRIOR WRITTEN CONSENT OF AN OFFICER OF RMI. POSSESSION OR RECEIPT OF THIS DOCUMENT DOES NOT CONVEY ANY RIGHT TO REPRODUCE IT OR DISCLOSE ITS CONTENTS, OR TO MANUFACTURE, USE, OR SELL ANYTHING BASED ON OR INCORPORATING THE CONCEPTS OR DETAILS DISCLOSED BY IT WITHOUT THE PRIOR WRITTEN PERMISSION OF RMI.
NOTES:
RATIONAL MACHINES INC.
6/22/82
TYPE_CHECKER
SHEET 1 OF 1
REV 0.3


MDATA_RCS
RAMPLANE0
RAMPLANE1
DIAGDRIVER
MDATA_DRVR
P0_RDLATCH
P1_RDLATCH
TS0_XCVR
TS1_XCVR
TAGSTORE0
TAGSTORE1
TS0TEST
TS1TEST
M_ADR_RCS
HASH_FUNC
TS_CONTROL
RAMADRGEN
REM_TIMING
DIAG_PROC
REM_CONTROL
THIS DOCUMENT CONTAINS DATA THAT IS DESIGNATED "TRADESECRET" BY RATIONAL MACHINES INC. AND IS NOT TO BE COPIED OR DISTRIBUTED WITHOUT THE PRIOR WRITTEN CONSENT OF AN OFFICER OF RMI. POSSESSION OR RECEIPT OF THIS DOCUMENT DOES NOT CONVEY ANY RIGHT TO REPRODUCE IT OR DISCLOSE ITS CONTENTS OR TO MANUFACTURE, USE, OR SELL ANYTHING BASED ON OR INCORPORATING THE CONCEPTS OR DETAILS DISCLOSED BY IT WITHOUT THE PRIOR WRITTEN PERMISSION OF RMI.
NOTES:
RATIONAL MACHINES INC.
6/6/82
MEMORY BOARD
SHEET 1 OF 1
REV 0.0


TYPE_DATA(0:63)
VALUE_DATA(0:63)
CHECK_BITS(0:8)
FIU_DATA(0:63)
TYPE_RCVR
VALUE_RCVR
FIU_RCVR
T_ASSN_REG
V_ASSN_REG
FIU_TIDATA(0:63)
FIU_VIDATA(0:63)
FIU_T_DRVR
FIU_V_DRVR
ERR_CHKGEN
CORR_ERR
MULTI_ERR
BADBITID(0:6)
MEMMONITOR
MEMCONT(0:119)
ROT_SEL(0:6)
BBIT_SEL(0:6)
ROTATOR
SIGN_BIT
UADDR(0:13)
ADDRESS(0:73)
MERGE_WHEN
ZEROBETECT
FIU_ZEROBO
MERGE_DATA
MDR_SIN
MDR_SOUT
LENGTH(0:6)
BADBITID(0:6)
BIT_OFFSET(0:6)
ROTDATA(0:63)
MERGE_V(0:63)
MERGER
MERGE_MASK
FIUCONTROLL
ROT_SEL(0:6)
BBIT_SEL(0:6)
COND_CODE
FIU_TODATA(0:63)
FIU_VODATA(0:63)
THIS DOCUMENT CONTAINS DATA THAT IS DESIGNATED "TRADESECRET" BY RATIONAL MACHINES INC. AND IS NOT TO BE COPIED OR DISTRIBUTED WITHOUT THE PRIOR WRITTEN CONSENT OF AN OFFICER OF RMI. POSSESSION OR RECEIPT OF THIS DOCUMENT DOES NOT CONVEY ANY RIGHT TO REPRODUCE IT OR DISCLOSE ITS CONTENTS, OR TO MANUFACTURE, USE, OR SELL ANYTHING BASED ON OR INCORPORATING THE CONCEPTS OR DETAILS DISCLOSED BY IT WITHOUT THE PRIOR WRITTEN PERMISSION OF RMI.
ROYCE0
4/29/82
RATIONAL MACHINES INC.
Field_Isolation_Unit
SHEET 1 OF 1
REV 0.1

REVISION HISTORY FOR FIU BOARD

1 4/20/82

Remove SIGN_BIT input from MERGE_TMUX and therefore delete the mux
* sign extension is only needed for 64 bits

Add zero detector to FIU_VODATA bus

Add VALUE/TYPE_DATA bus receivers for ERCC
* allows FIU flexibility during LOAD_WDR and READ_RDR ops

Renamed ROTDAT_REG to MERGE_DREG
* ROTDAT_REG was being confused with RDR (READ_DREG)

Make MERGE_DREG diagnostically scannable

Add connection from MERGE_DREG to FIU_VODATA bus
* allows MERGE_DREG to be read and written in same cycle
* increases diagnostic visibility

Change connection to ADDRESS bus from input to bidirectional

Add detail to control (SBIT_SEL(0:6), BADBITID(0:6), COND_CODE,
FIU_EQZERO)

BLOCK COPY FLOW

CASE 1

R0	-	T(W1)	->	
R1	<- V(W1)	T(W2)	->	
R2	<- V(W2)	-	-	

BUSSES / REGISTERS

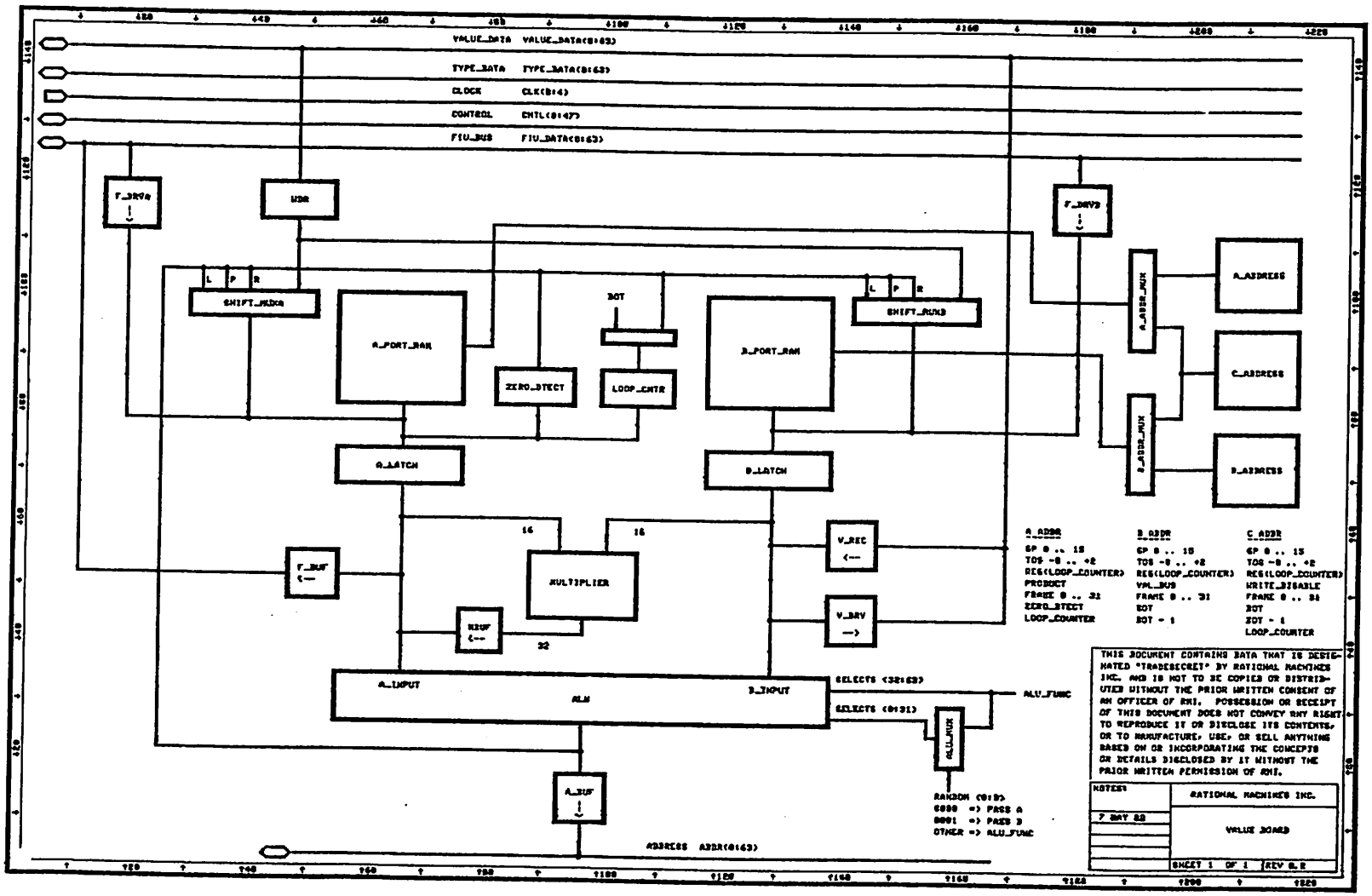
		TYPE	VALUE	FIU	TAR	VAR	MDR
1:	LOAD_MAR(R1(V)),	T(R0)	V(R0)	-	-	-	-
	START READ,			T(W1)	-	-	-
	EXTRACT((T,V),T(W1) -> FIU);			LOAD			
2:	CONTINUE,	-	V(R0)	-	-	-	-
	V(R0) -> VAR;			-	-	V(R0)	-
3:	READ_RDR,	T(R1)	V(R1)	-	-	V(R0)	-
	T(R1) -> TAR,		LOAD	-	T(R1)	V(R0)	T(W2)
	EXTRACT((T,V),T(W2) -> MDR);						
4:	READ_RDR,	T(R2)	V(R2)	-	T(R1)	V(R0)	T(W2)
	LOAD_MAR(W1(V)),	LOAD	LOAD	-	T(R1)	V(W1)	T(W2)
	START WRITE,						
	EXTRACT((TA,VA),V(W1) -> VAR);						
5:	LOAD_WDR(T,VA);	T(W1)	V(W1)	-	-	V(W1)	T(W2)
				-	-	V(W1)	T(W2)
6:	LOAD_MAR(W2(V)),	T(R2)	V(R1)	-	-	-	T(W2)
	START WRITE,			V(W2)	-	-	T(W2)
	EXTRACT((T,V),V(W2) -> FIU);			LOAD			
7:	LOAD_WDR(TF,V),	T(W2)	V(W2)	T(W2)	-	-	T(W2)
	MDR -> FIU_VO -> T;			T(W2)	-	-	T(W2)

Duplicate source, minor DBT

CASE 2

	-	-	->
R1	<- T(W1)	V(W1)	->
R2	<- T(W2)	V(W2)	-

		BUSSES			REGISTERS		
		TYPE	VALUE	FIU	TAR	VAR	MDR
1:	LOAD_MAR(R1(V)),	-	-	-	-	-	-
	START READ;			-	-	-	-
2:	CONTINUE,	-	V(RO)	-	-	-	-
	V(RO) -> VAR;			-	-	V(RO)	-
3:	READ_RDR,	T(R1)	V(R1)	-	-	V(RO)	-
	T(R1) -> TAR,		LOAD	T(W1)	T(R1)	V(RO)	-
	EXTRACT((T,VA),T(W1) -> FIU);			LOAD			
4:	READ_RDR,	T(R2)	V(R2)	V(R1)	T(R1)	-	-
	V(R1) -> FIU -> VAR,	LOAD	LOAD	V(R1)	T(R1)	V(R1)	V(W2)
	EXTRACT((T,V),V(W2) -> MDR);						
	LOAD_MAR(W1(V)),	-	-	-	T(R1)	V(R1)	V(W2)
	START WRITE,			V(W1)	T(R1)	V(R1)	V(W2)
	EXTRACT((TA,VA),V(W1) -> FIU);			LOAD			
6:	LOAD_WDR(T,V),	T(W1)	V(W1)	T(R2)	-	V(R1)	V(W2)
	CONTINUE,			T(R2)	T(W2)	V(R1)	V(W2)
	EXTRACT((TF,VA),T(W2) -> TAR);						
7:	LOAD_WDR(TA,VF),	T(W2)	V(W2)	V(W2)	T(W2)	-	V(W2)
	MDR -> FIU_VO -> V;			V(W2)	T(W2)	-	V(W2)



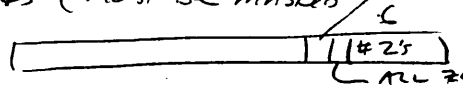
FIU Bus Mux'ed to RF on UAL, MRE is illegal

cycle

n-1 COUNT # LEADING 0's

n USE # LEADING 0's (MUST BE MASKED

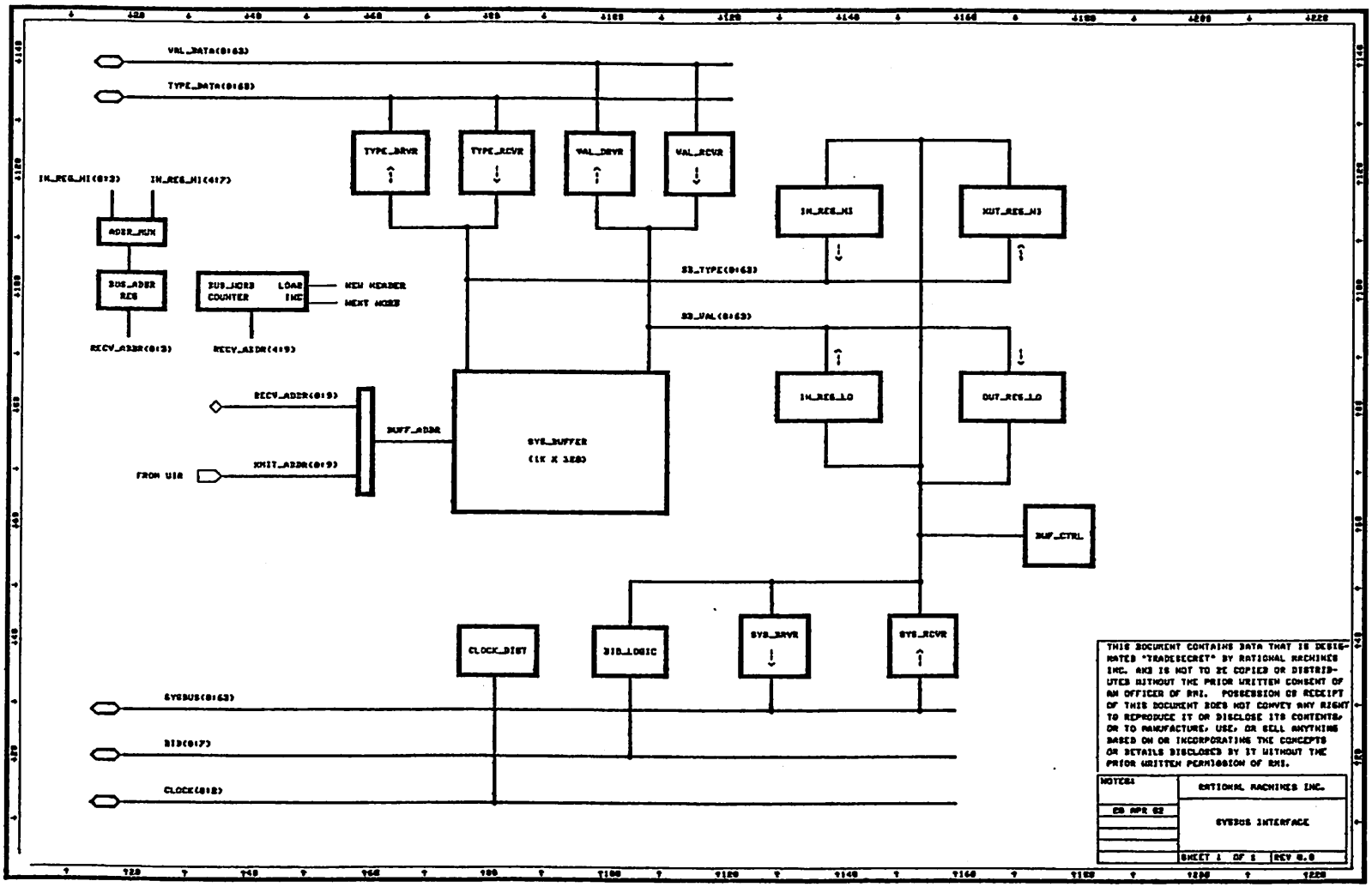
(LAZY MOD)

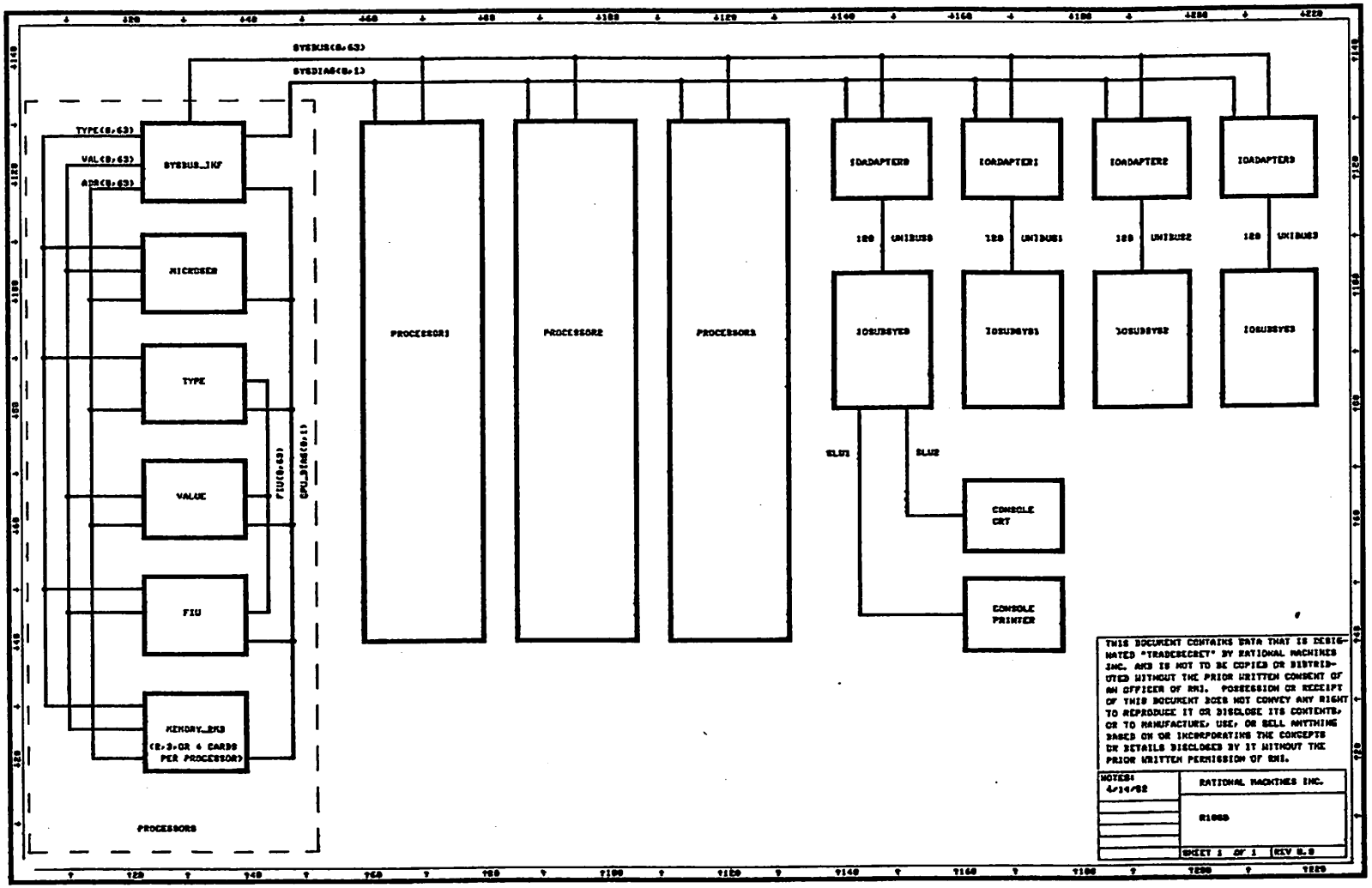


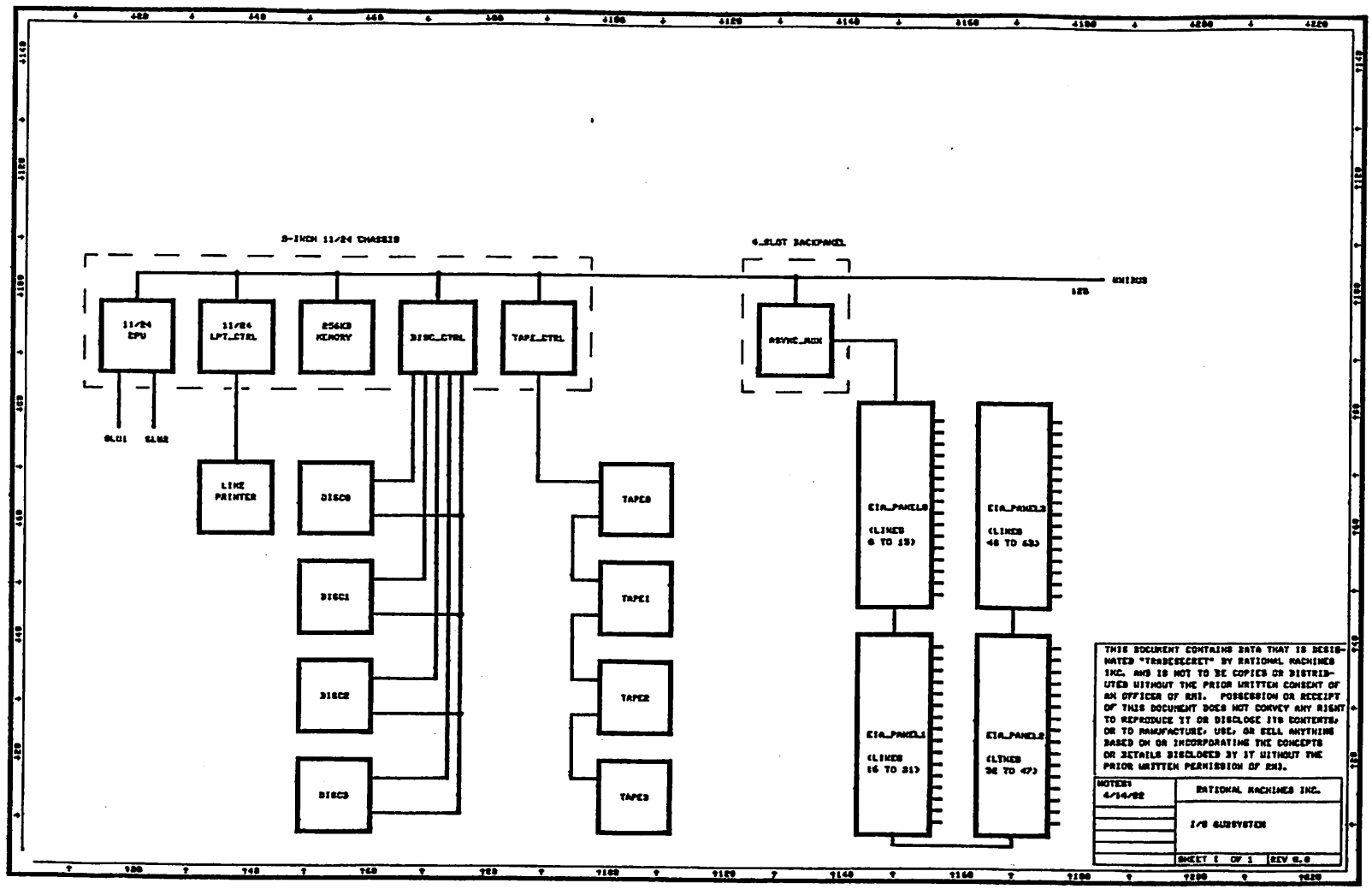
UPPER 33 ARE SAME SIGN

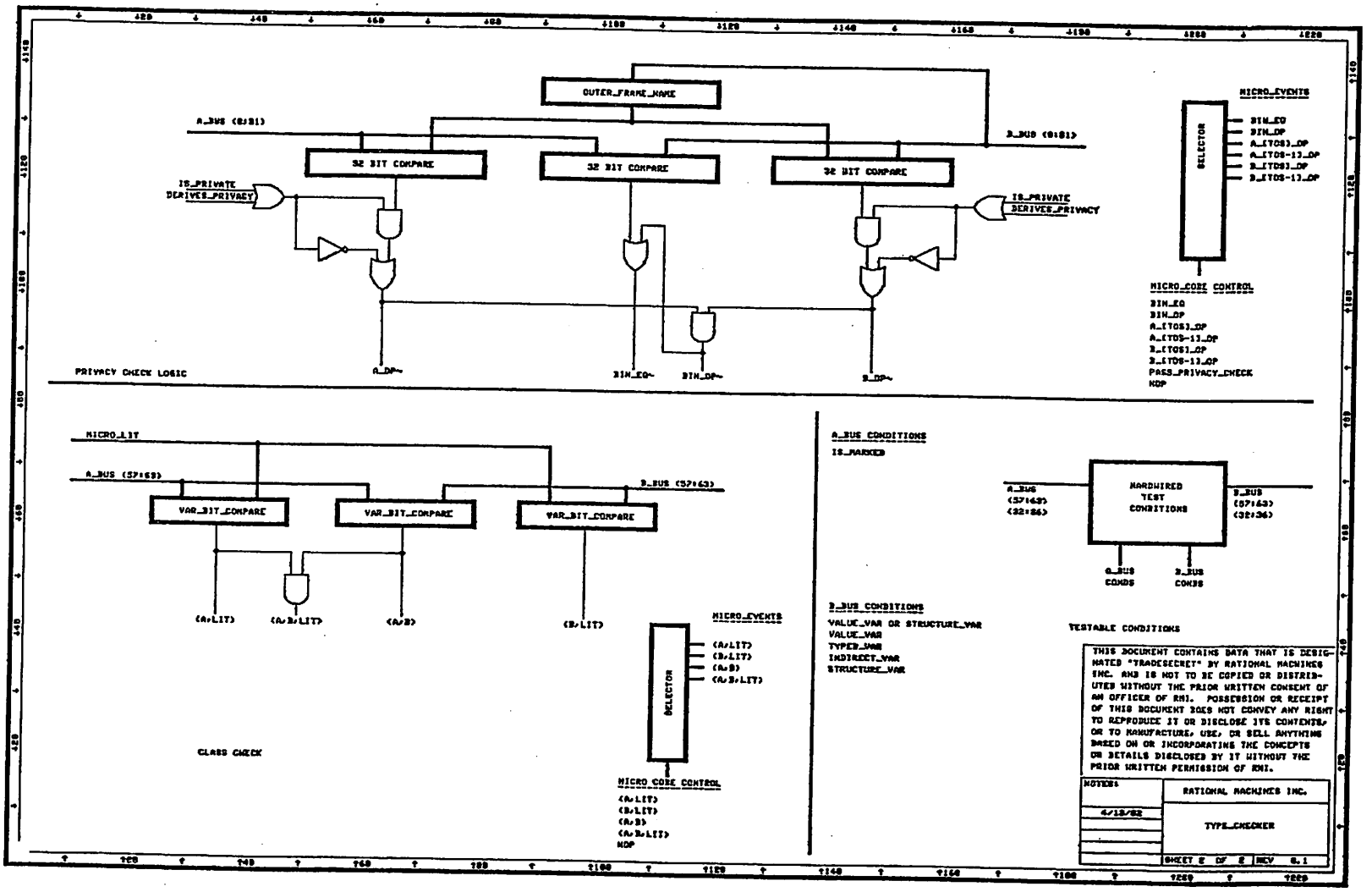
ALL ZERO

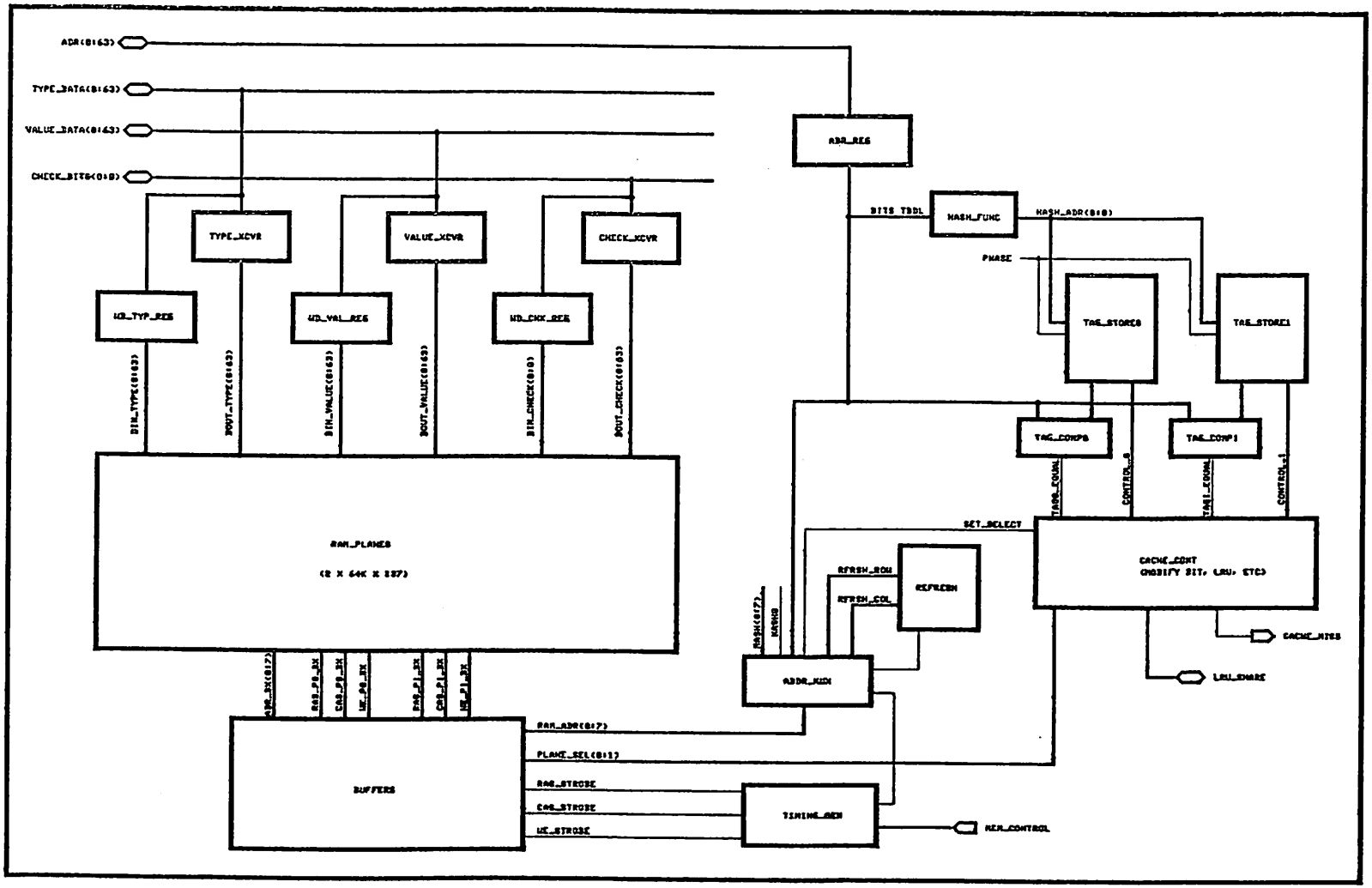
ALSO CONDITIONS

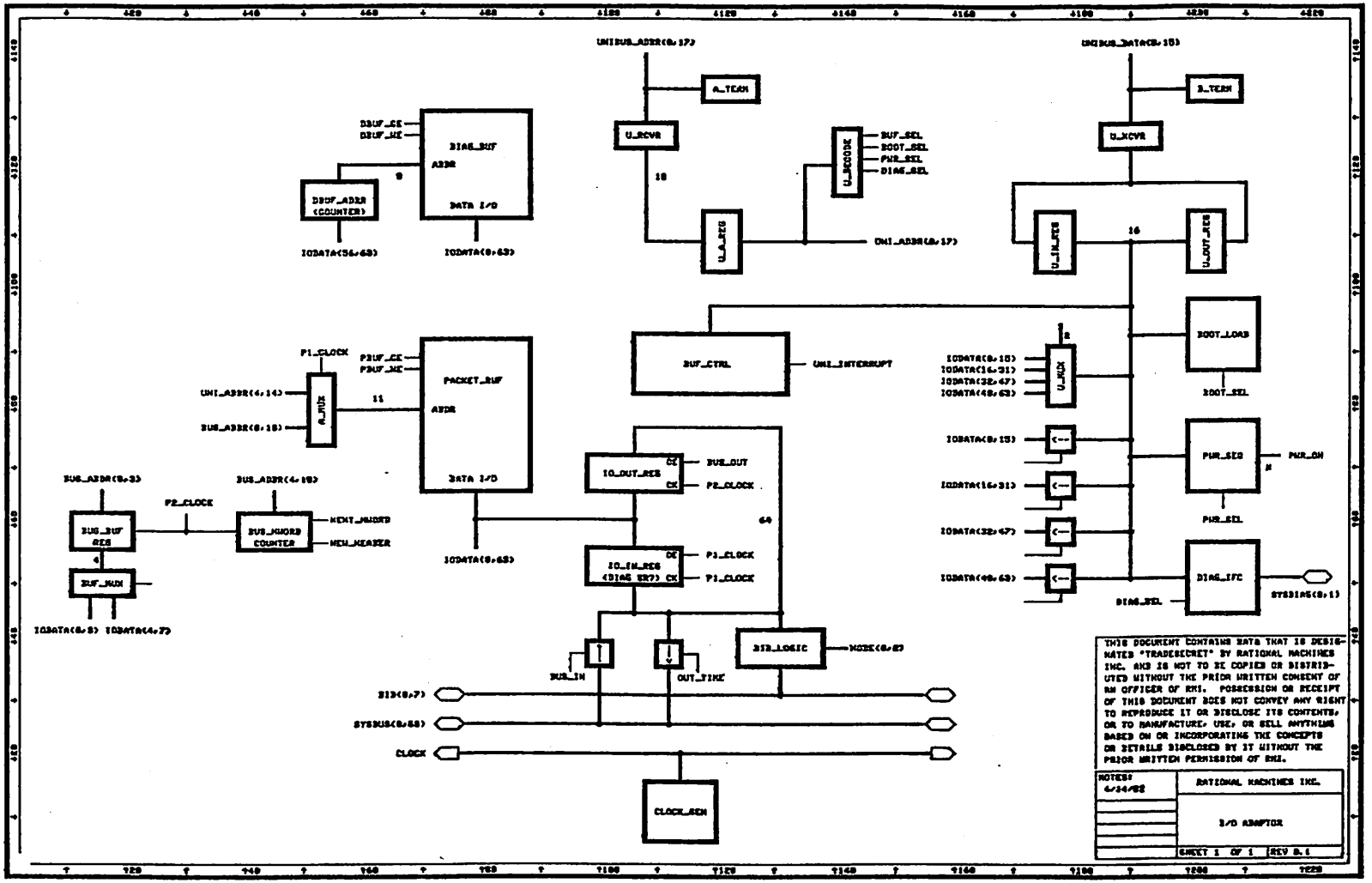

VAL_DATA(0:63)
TYPE_DATA(0:63)
TYPE_DRVR
TYPE_RCVR
VAL_DRVR
VAL_RCVR
IN_REG_HI(0:3)
IN_REG_HI(4:7)
ADER_RUN
SUS_ADER REG
BUS_HDR5 COUNTER
LOAD
INC
NEW HEADER
NEXT HDR5
RECV_ADDR(0:3)
RECV_ADDR(4:9)
IN_REG_HI
OUT_REG_HI
SB_TYPE(0:63)
SB_VAL(0:63)
IN_REG_LO
OUT_REG_LO
RECV_ADDR(0:9)
XMIT_ADDR(0:9)
FROM UIB
BUFF_ADDR
SYS_BUFFER
(1K X 128)
BUF_CTRL
CLOCK_DIST
BID_LOGIC
SYS_DRVR
SYS_RCVR
SYSBUS(0:63)
BID(0:7)
CLOCK(0:2)
THIS DOCUMENT CONTAINS DATA THAT IS DESIGNATED "TRADESECRET" BY RATIONAL MACHINES INC. AND IS NOT TO BE COPIED OR DISTRIBUTED WITHOUT THE PRIOR WRITTEN CONSENT OF AN OFFICER OF RMI. POSSESSION OR RECEIPT OF THIS DOCUMENT DOES NOT CONVEY ANY RIGHT TO REPRODUCE IT OR DISCLOSE ITS CONTENTS, OR TO MANUFACTURE, USE, OR SELL ANYTHING BASED ON OR INCORPORATING THE CONCEPTS OR DETAILS DISCLOSED BY IT WITHOUT THE PRIOR WRITTEN PERMISSION OF RMI.
NOTICE:
RATIONAL MACHINES INC.
28 APR 82
SYSBUS INTERFACE
SHEET 1 OF 1
REV 0.0

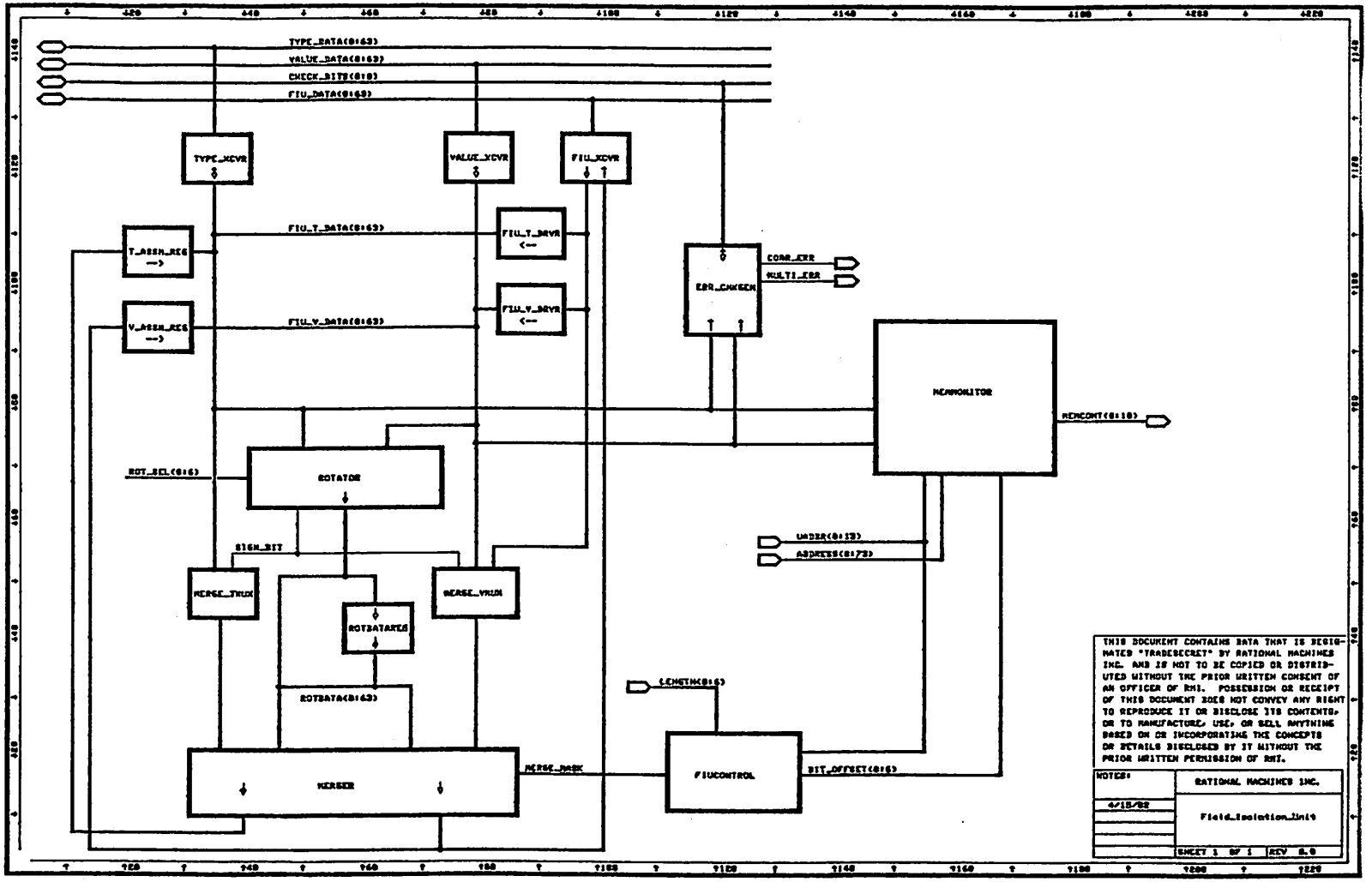

SYSBUS(0:63)
SYSDIAG(0:1)
TYPE(0:63)
VAL(0:63)
ADR(0:63)
SYSBUS_INF
NICKDEB
TYPE
VALUE
FIU
MEMORY_BRD
(2,3,OR 4 CARDS
PER PROCESSOR)
PROCESSORS
FIU(0:63)
CPU_DIAG(0:1)
PROCESSOR3
PROCESSOR2
PROCESSOR1
IOADAPTER0
IOADAPTER1
IOADAPTER2
IOADAPTER3
128
UNIBUS0
UNIBUS1
UNIBUS2
UNIBUS3
IOSUBSYS0
IOSUBSYS1
IOSUBSYS2
IOSUBSYS3
SLU1
SLU2
CONSOLE CRT
CONSOLE PRINTER
THIS DOCUMENT CONTAINS DATA THAT IS DESIGNATED "TRADESECRET" BY RATIONAL MACHINES INC. AND IS NOT TO BE COPIED OR DISTRIBUTED WITHOUT THE PRIOR WRITTEN CONSENT OF AN OFFICER OF RMI. POSSESSION OR RECEIPT OF THIS DOCUMENT DOES NOT CONVEY ANY RIGHT TO REPRODUCE IT OR DISCLOSE ITS CONTENTS, OR TO MANUFACTURE, USE, OR SELL ANYTHING BASED ON OR INCORPORATING THE CONCEPTS OR DETAILS DISCLOSED BY IT WITHOUT THE PRIOR WRITTEN PERMISSION OF RMI.
NOTES:
4/14/82
RATIONAL MACHINES INC.
R1000
SHEET 1 OF 1
REV 0.0

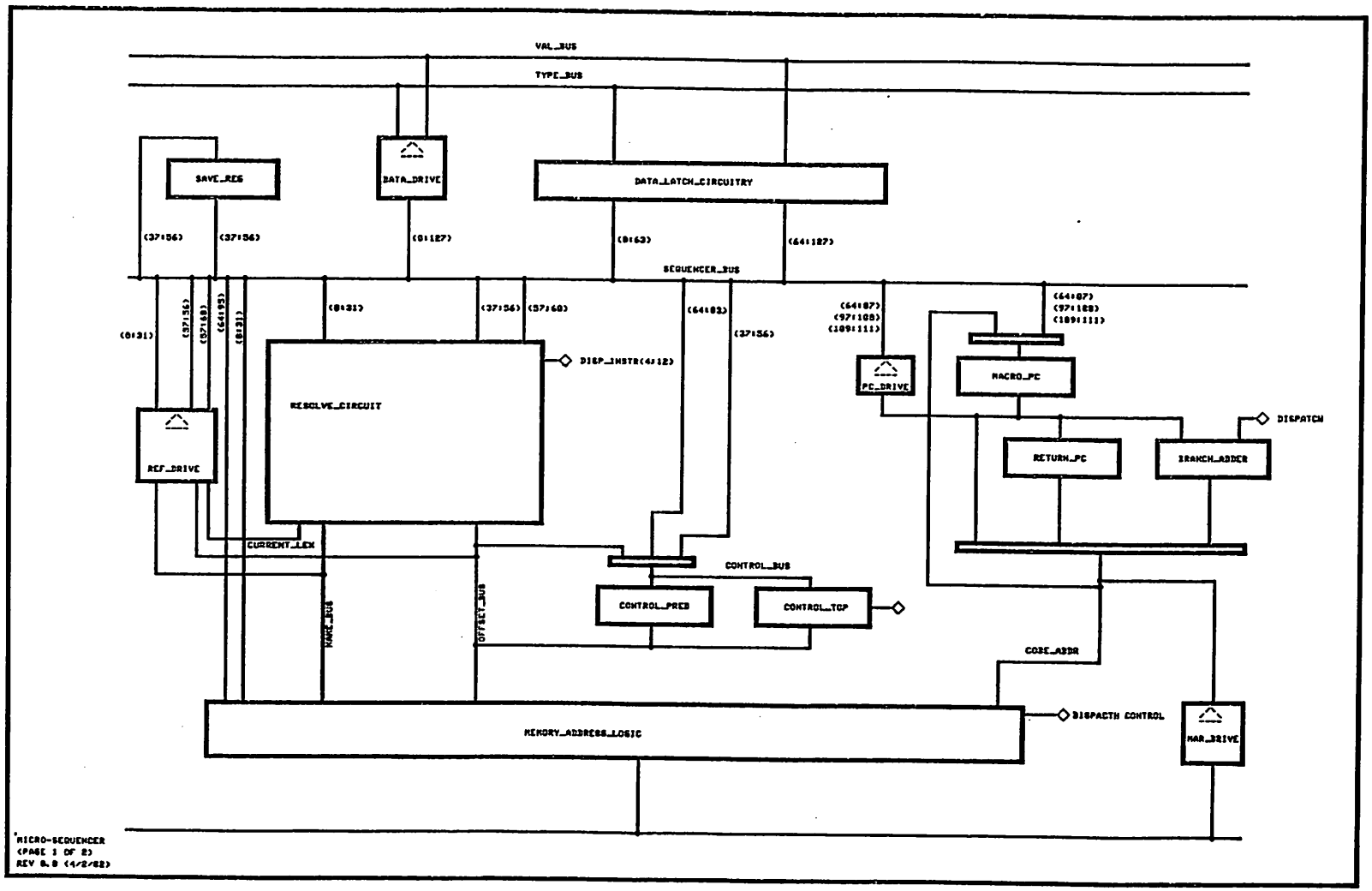

8-INCH 11/24 CHASSIS
6-SLOT BACKPANEL
11/24 CPU
11/24 LPT_CTRL
256KB MEMORY
DISC_CTRL
TAPE_CTRL
ASYNC_MUX
LINE PRINTER
DISC0
DISC1
DISC2
DISC3
TAPE0
TAPE1
TAPE2
TAPE3
EIA_PANEL0 (LINES 0 TO 15)
EIA_PANEL3 (LINES 48 TO 63)
EIA_PANEL1 (LINES 16 TO 31)
EIA_PANEL2 (LINES 32 TO 47)
THIS DOCUMENT CONTAINS DATA THAT IS DESIGNATED "TRADESECRET" BY NATIONAL MACHINES INC. AND IS NOT TO BE COPIES OR DISTRIBUTED WITHOUT THE PRIOR WRITTEN CONSENT OF AN OFFICER OF NMI. POSSESSION OR RECEIPT OF THIS DOCUMENT DOES NOT CONVEY ANY RIGHT TO REPRODUCE IT OR DISCLOSE ITS CONTENTS, OR TO MANUFACTURE, USE, OR SELL ANYTHING BASED ON OR INCORPORATING THE CONCEPTS OR DETAILS DISCLOSED BY IT WITHOUT THE PRIOR WRITTEN PERMISSION OF NMI.
NOTES:
4/14/82
NATIONAL MACHINES INC.
I/O SUBSYSTEM
SHEET 1 OF 1
REV 0.0

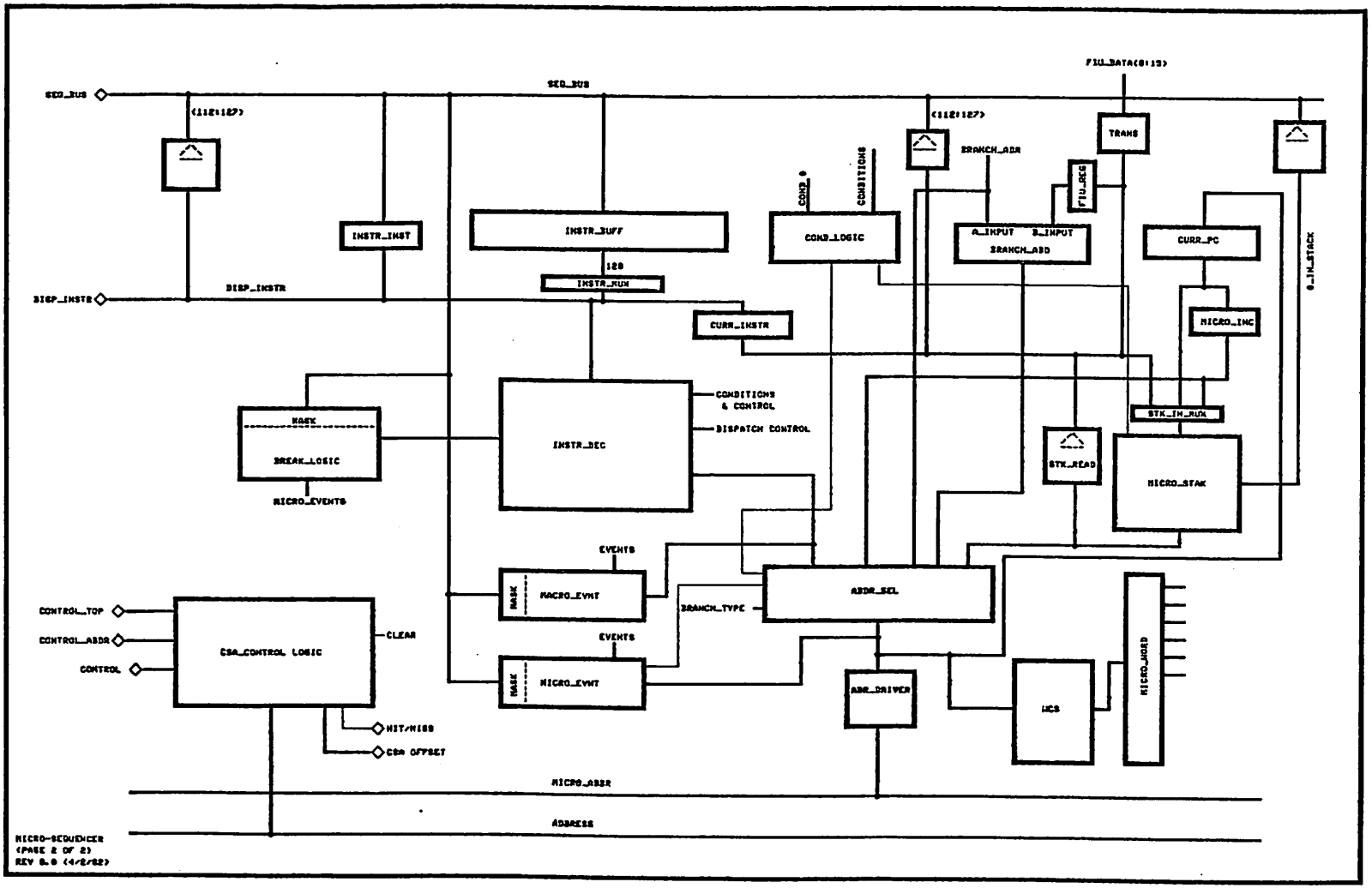

OUTER_FRAME_NAME
A_BUS (0:31)
B_BUS (0:31)
32 BIT COMPARE
32 BIT COMPARE
32 BIT COMPARE
IS_PRIVATE DERIVES_PRIVACY
IS_PRIVATE DERIVES_PRIVACY
A_OP~
BIN_EQ~
BIN_OP~
B_OP~
PRIVACY CHECK LOGIC
MICRO_EVENTS
SELECTOR
BIN_EQ
BIN_OP
A_(TOS)_OP
A_(TOS-1)_OP
B_(TOS)_OP
B_(TOS-1)_OP
MICRO_CODE CONTROL
BIN_EQ
BIN_OP
A_(TOS)_OP
A_(TOS-1)_OP
B_(TOS)_OP
B_(TOS-1)_OP
PASS_PRIVACY_CHECK
NOP
MICRO_LIT
A_BUS (57:63)
B_BUS (57:63)
VAR_BIT_COMPARE
VAR_BIT_COMPARE
VAR_BIT_COMPARE
(A:LIT)
(A:B:LIT)
(A:B)
(B:LIT)
CLASS CHECK
MICRO_EVENTS
SELECTOR
(A:LIT)
(B:LIT)
(A:B)
(A:B:LIT)
MICRO_CODE CONTROL
(A:LIT)
(B:LIT)
(A:B)
(A:B:LIT)
NOP
A_BUS CONDITIONS
IS_MARKED
A_BUS (57:63) (32:56)
HARDWIRED TEST CONDITIONS
B_BUS (57:63) (32:56)
A_BUS CONDS
B_BUS CONDS
B_BUS CONDITIONS
VALUE_VAR OR STRUCTURE_VAR
VALUE_VAR
TYPED_VAR
INDIRECT_VAR
STRUCTURE_VAR
TESTABLE CONDITIONS
THIS DOCUMENT CONTAINS DATA THAT IS DESIGNATED "TRADESECRET" BY RATIONAL MACHINES INC. AND IS NOT TO BE COPIED OR DISTRIBUTED WITHOUT THE PRIOR WRITTEN CONSENT OF AN OFFICER OF RMI. POSSESSION OR RECEIPT OF THIS DOCUMENT DOES NOT CONVEY ANY RIGHT TO REPRODUCE IT OR DISCLOSE ITS CONTENTS, OR TO MANUFACTURE, USE, OR SELL ANYTHING BASED ON OR INCORPORATING THE CONCEPTS OR DETAILS DISCLOSED BY IT WITHOUT THE PRIOR WRITTEN PERMISSION OF RMI.
NOTES:
RATIONAL MACHINES INC.
4/13/82
TYPE_CHECKER
SHEET 2 OF 2 REV 0.1

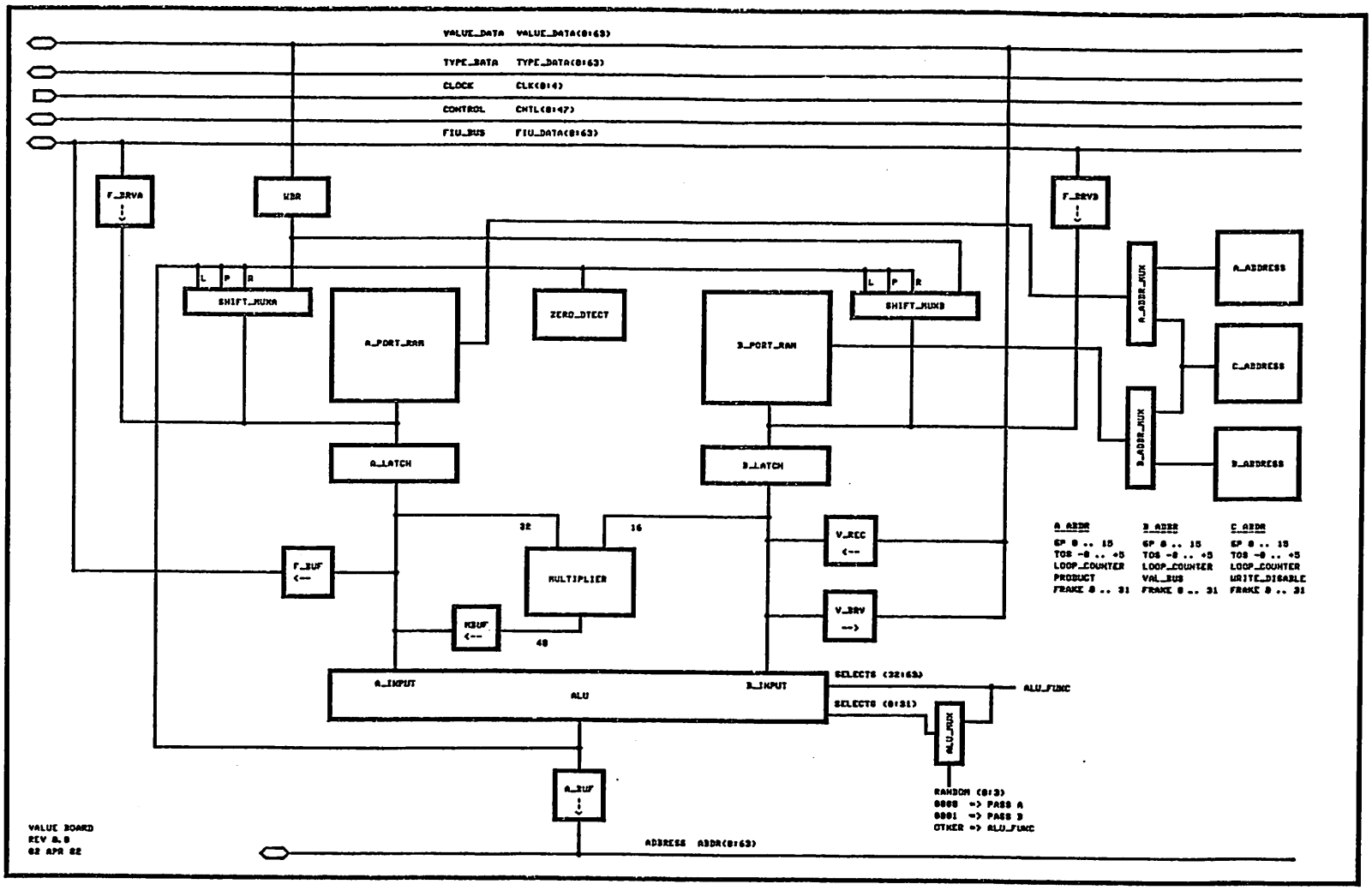

ADR(0:63)
TYPE_DATA(0:63)
VALUE_DATA(0:63)
CHECK_BITS(0:8)
ADR_REG
BITS TBDL
HASH_FUNC
HASH_ADR(0:8)
PHASE
TYPE_XCVR
VALUE_XCVR
CHECK_XCVR
WD_TYP_REG
WD_VAL_REG
WD_CHK_REG
TAG_STORE0
TAG_STORE1
DIN_TYPE(0:63)
DOUT_TYPE(0:63)
DIN_VAL(0:63)
DOUT_VALUE(0:63)
DIN_CHECK(0:8)
DOUT_CHECK(0:63)
TAG_COMP0
TAG_COMP1
RAM_PLANES
(2 X 64K X 137)
SET_SELECT
CACHE_CONT
(DIRTY BIT, LRU, ETC)
REFRESH
RFRSH_ROW
RFRSH_COL
CACHE_HITS
LRU_SHARE
ADDR_MUX
BUFFERS
RAM_ADR(0:7)
PLANE_SEL(0:1)
RAS_STROBE
CAS_STROBE
WE_STROBE
TIMING_GEN
MEM_CONTROL

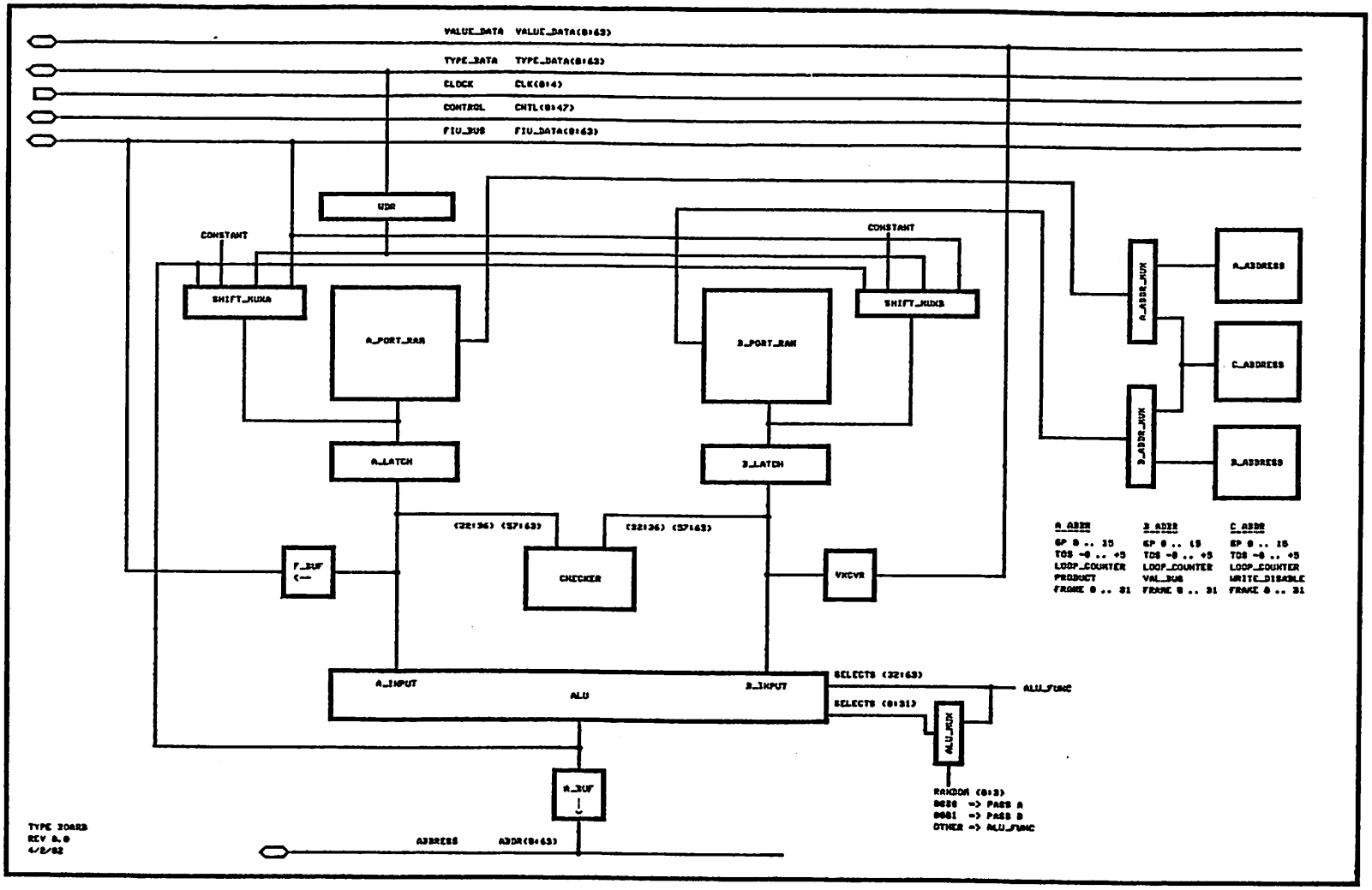

UNIBUS_ADDR(0:17)
A_TERM
U_RCVR
18
U_A_REG
U_DECODE
BUF_SEL
BOOT_SEL
PWR_SEL
DIAG_SEL
UNI_ADDR(0:17)
DBUF_CE
DBUF_WE
DIAG_BUF
ADDR
DATA I/O
9
DBUF_ADDR (COUNTER)
IODATA(56:63)
IODATA(0:63)
UNIBUS_DATA(0:15)
B_TERM
U_RCVR
U_IN_REG
16
U_OUT_REG
BOOT_LOAD
BOOT_SEL
BUF_CTRL
UNI_INTERRUPT
U_MUX
IODATA(0:15)
IODATA(16:31)
IODATA(32:47)
IODATA(48:63)
P1_CLOCK
PBUF_CE
PBUF_WE
PACKET_BUF
ADDR
DATA I/O
A_MUX
UNI_ADDR(4:14)
BUS_ADDR(0:15)
11
IODATA(0:63)
IO_OUT_REG
CE
CK
BUS_OUT
P2_CLOCK
64
IO_IN_REG (DIAG REG)
CE
CK
P1_CLOCK
P1_CLOCK
PWR_REG
PWR_ON
PWR_SEL
DIAG_IFC
SYSBUS(0:1)
DIAG_SEL
BUS_ADDR(0:3)
BUS_ADDR(4:18)
P2_CLOCK
BUS_BUF REG
4
BUF_NUM
IODATA(0:3)
IODATA(4:7)
BUS_NUMBER COUNTER
NEXT_NUMBER
NEW_HEADER
BUS_IN
OUT_PIKE
BIB_LOGIC
MODE(0:2)
BI3(0:7)
SYSBUS(0:63)
CLOCK
CLOCK_REG
THIS DOCUMENT CONTAINS DATA THAT IS DESIGNATED "TRADESECRET" BY RATIONAL MACHINES INC. AND IS NOT TO BE COPIED OR DISTRIBUTED WITHOUT THE PRIOR WRITTEN CONSENT OF AN OFFICER OF RMI. POSSESSION OR RECEIPT OF THIS DOCUMENT DOES NOT CONVEY ANY RIGHT TO REPRODUCE IT OR DISCLOSE ITS CONTENTS, OR TO MANUFACTURE, USE, OR SELL ANYTHING BASED ON OR INCORPORATING THE CONCEPTS OR DETAILS DISCLOSED BY IT WITHOUT THE PRIOR WRITTEN PERMISSION OF RMI.
NOTES:
4/24/82
RATIONAL MACHINES INC.
I/O ADAPTOR
SHEET 1 OF 1
REV A.1

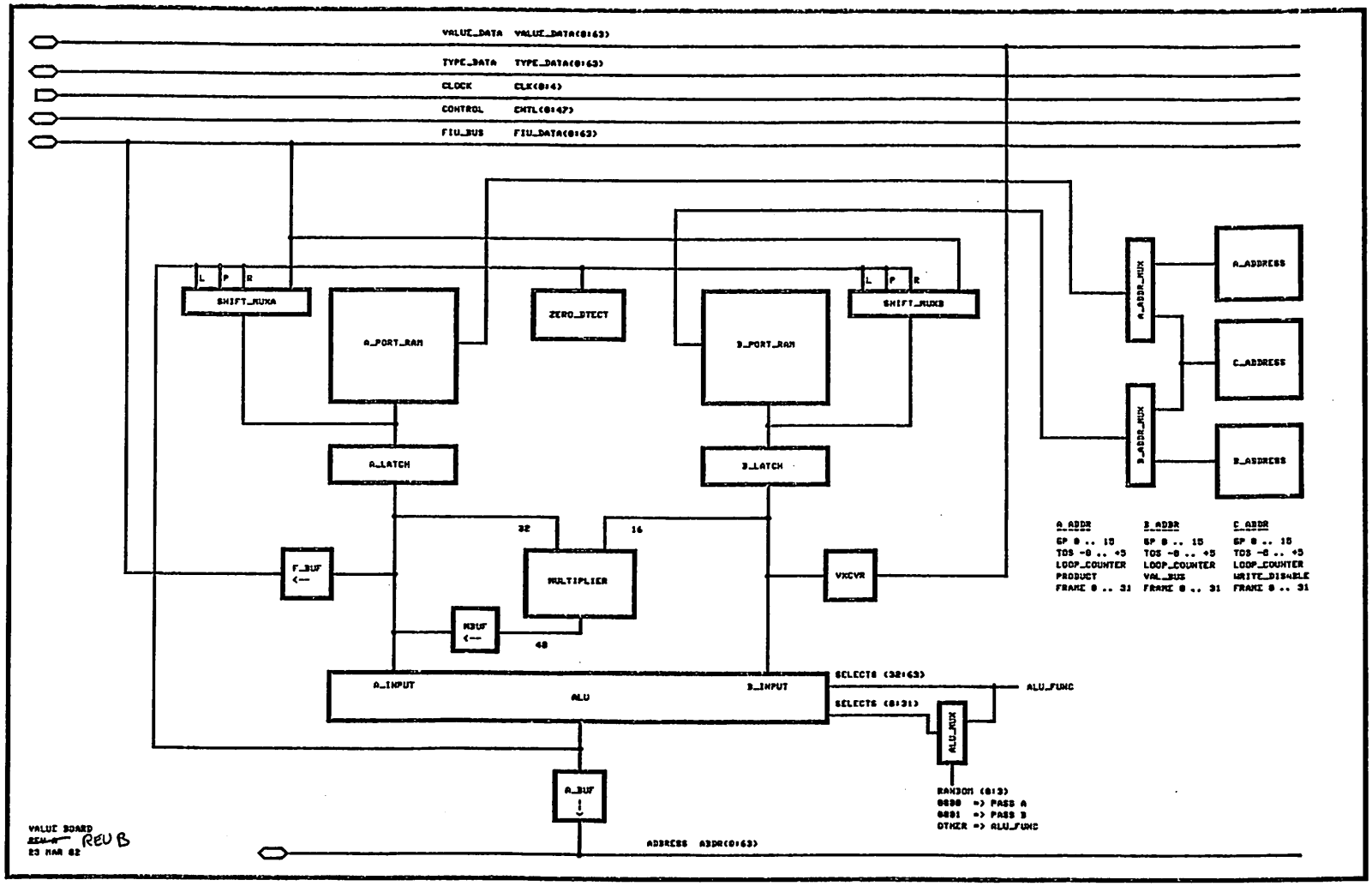

TYPE_DATA(0:63)
VALUE_DATA(0:63)
CHECK_BITS(0:8)
FIU_DATA(0:63)
TYPE_XCVR
VALUE_XCVR
FIU_XCVR
T_ASSN_REG
V_ASSN_REG
FIU_T_DRVR
FIU_V_DRVR
FIU_T_DATA(0:63)
FIU_V_DATA(0:63)
ERR_CHKGEN
CORR_ERR
MULTI_ERR
MEMMONITOR
MEMCONT(0:18)
ROT_SEL(0:6)
ROTATOR
SIGN_BIT
MERGE_TRUX
MERGE_VALX
ROTDATA(0:63)
UNDER(0:2)
ADDRESS(0:27)
CHKSYNC(0:6)
MERGER
MERGE_MASK
FIUCONTROL
BIT_OFFSET(0:6)
THIS DOCUMENT CONTAINS DATA THAT IS DESIGNATED "TRADESECRET" BY RATIONAL MACHINES INC. AND IS NOT TO BE COPIED OR DISTRIBUTED WITHOUT THE PRIOR WRITTEN CONSENT OF AN OFFICER OF RMI. POSSESSION OR RECEIPT OF THIS DOCUMENT DOES NOT CONVEY ANY RIGHT TO REPRODUCE IT OR DISCLOSE ITS CONTENTS, OR TO MANUFACTURE, USE, OR SELL ANYTHING BASED ON OR INCORPORATING THE CONCEPTS OR DETAILS DISCLOSED BY IT WITHOUT THE PRIOR WRITTEN PERMISSION OF RMI.
NOTES:
RATIONAL MACHINES INC.
4/15/82
Field_Isolation_Unit
SHEET 1 OF 1
REV 0.0


VAL_BUS
TYPE_BUS
SAVE_REG
DATA_DRIVE
DATA_LATCH_CIRCUITRY
(37:56)
(37:56)
(0:127)
(0:63)
(64:127)
SEQUENCER_BUS
(0:31)
(37:56)
(37:168)
(64:95)
(0:31)
(0:31)
(37:56)
(57:60)
(64:82)
(37:56)
(64:87)
(97:108)
(109:111)
(64:87)
(97:108)
(109:111)
DISP_INSTR(4:12)
RESOLVE_CIRCUIT
REF_DRIVE
PC_DRIVE
MACRO_PC
DISPATCH
RETURN_PC
BRANCH_ADDER
CURRENT_LEX
CONTROL_BUS
CONTROL_PREB
CONTROL_TOP
NAME_BUS
OFFSET_BUS
CODE_ADDR
DISPACTH CONTROL
MEMORY_ADDRESS_LOGIC
MAR_DRIVE
MICRO-SEQUENCER
(PAGE 1 OF 2)
REV 8.8 (4/2/82)


SEQ_BUS
(112:127)
SEQ_BUS
(112:127)
FIU_DATA(0:15)
TRANS
BRANCH_ADR
FIU_REG
INSTR_INST
INSTR_BUFF
COND
CONDITIONS
COND_LOGIC
A_INPUT
B_INPUT
BRANCH_ADD
CURR_PC
128
INSTR_RUN
DISP_INSTR
DISP_INSTR
CURR_INSTR
MICRO_INC
0_IN_STACK
MASK
BREAK_LOGIC
MICRO_EVENTS
INSTR_DEC
CONDITIONS & CONTROL
DISPATCH CONTROL
STK_IN_RUN
STK_READ
MICRO_STAK
EVENTS
MASK
MACRO_EVNT
BRANCH_TYPE
ADDR_SEL
CONTROL_TOP
CONTROL_ADDR
CONTROL
CSA_CONTROL LOGIC
CLEAR
EVENTS
MASK
MICRO_EVNT
ADDR_DRIVER
UCS
MICRO_WORD
HIT/MISS
CSA OFFSET
MICRO_ADDR
ADDRESS
MICRO-SEQUENCER
(PAGE 2 OF 2)
REV 5.0 (4/2/82)


VALUE_DATA VALUE_DATA(0:63)
TYPE_DATA TYPE_DATA(0:63)
CLOCK CLK(0:4)
CONTROL CNTL(0:47)
FIU_BUS FIU_DATA(0:63)
F_SRVA
KBR
F_SRVB
L P R
SHIFT_MUXA
ZERO_DTECT
SHIFT_MUXB
A_PORT_RAM
B_PORT_RAM
A_ADDR_MUX
B_ADDR_MUX
A_ADDRESS
C_ADDRESS
B_ADDRESS
A_LATCH
B_LATCH
32
16
F_BUF
MULTIPLIER
V_REC
V_SRV
KBUF
48
A_INPUT
ALU
B_INPUT
SELECTS (32:63)
SELECTS (0:31)
ALU_FUNC
ALU_MUX
A_BUF
RANDOM (0:3)
0000 => PASS A
0001 => PASS B
OTHER => ALU_FUNC
ADDRESS ADDR(0:63)
A ADDR
GP 0 .. 15
TOS -8 .. +5
LOOP_COUNTER
PRODUCT
FRAME 0 .. 31
B ADDR
GP 0 .. 15
TOS -8 .. +5
LOOP_COUNTER
VAL_BUS
FRAME 0 .. 31
C ADDR
GP 0 .. 15
TOS -8 .. +5
LOOP_COUNTER
WRITE_DISABLE
FRAME 0 .. 31
VALUE BOARD
REV 0.0
02 APR 02


VALUE_DATA VALUE_DATA(0:63)
TYPE_DATA TYPE_DATA(0:63)
CLOCK CLK(0:4)
CONTROL CNTL(0:47)
FIU_BUS FIU_DATA(0:63)
UDR
CONSTANT
CONSTANT
SHIFT_MUXA
SHIFT_MUXB
A_PORT_RAM
B_PORT_RAM
A_ADDR_MUX
B_ADDR_MUX
A_ADDRESS
C_ADDRESS
B_ADDRESS
A_LATCH
B_LATCH
(32:36) (57:63)
(32:36) (57:63)
F_BUF
CHECKER
VXCVR
A_INPUT
ALU
B_INPUT
SELECTS (32:63)
SELECTS (0:31)
ALU_FUNC
ALU_MUX
A_BUF
RANDOM (0:3)
0000 => PASS A
0001 => PASS B
OTHER => ALU_FUNC
ADDRESS ADDR(0:63)
TYPE BOARD
REV A.0
4/2/82
A ADDR
GP 0 .. 15
TOS -0 .. +5
LOOP_COUNTER
PRODUCT
FRAME 0 .. 31
B ADDR
GP 0 .. 15
TOS -0 .. +5
LOOP_COUNTER
VAL_BUS
FRAME 0 .. 31
C ADDR
GP 0 .. 15
TOS -0 .. +5
LOOP_COUNTER
WRITE_DISABLE
FRAME 0 .. 31


VALUE_DATA VALUE_DATA(0:63)
TYPE_DATA TYPE_DATA(0:63)
CLOCK CLK(0:4)
CONTROL CNTL(0:47)
FIU_BUS FIU_DATA(0:63)
L P R
SHIFT_MUXA
A_PORT_RAM
ZERO_DTECT
B_PORT_RAM
SHIFT_MUXB
A_ADDR_MUX
B_ADDR_MUX
A_ADDRESS
C_ADDRESS
B_ADDRESS
A_LATCH
B_LATCH
32
16
F_BUF
MULTIPLIER
VXCVR
M3UF
48
A_INPUT
ALU
B_INPUT
SELECT8 (32:63)
SELECT8 (0:31)
ALU_FUNC
ALU_MUX
A_BUF
RANDOM (0:3)
0000 => PASS A
0001 => PASS B
OTHER => ALU_FUNC
ADDRESS ADDR(0:63)
A ADDR
GP 0 .. 15
TOS -8 .. +5
LOOP_COUNTER
PRODUCT
FRAME 0 .. 31
B ADDR
GP 0 .. 15
TOS -8 .. +5
LOOP_COUNTER
VAL_BUS
FRAME 0 .. 31
C ADDR
GP 0 .. 15
TOS -8 .. +5
LOOP_COUNTER
WRITE_DISABLE
FRAME 0 .. 31
VALUE BOARD
REV B
23 MAR 82

This is a list of micro events as the stand today. They are divided to several different classes.

	condition	E/L	mask	specify	priority
MEMORY ERRORS					
page fault	X	E	X		2
ECC error		E			3
page mode error		E			4
TYPE CHECK ERRORS					
unary class error	X			X	5
binary class error	X			X	6
privacy check error	X	E		X	
ARITHMETIC					
val_overflow (64 bits)	X				
type_overflow (64 bits)	X				
max_bound error	X			X	
min_bound error	X			X	
EXTERNALS					
timer	X		X		
diagnostic_interrupt	X		X		
power failure	X		X		1
sysbus	X		X		

Conditions are events that can also be tested as conditions. Maskable events can be turned off, by setting the corresponding bit in a register on the micro-sequencer. Specifiable events are enabled/disabled on a per micro-instruction basis. Some events have the indicated priority levels (1 is the highest priority).

Events which are early (E) cause the execution of the current micro-instruction to be stopped. The next micro-instruction executed is a NOP, and the following micro-instruction is the first micro-instruction of the appropriate event handler (Each event maps to a unique address). Events which are late allow the current micro-instruction to complete and inhibit the completion of the next micro-instruction. The instruction following the inhibited micro-instruction is the first micro-instruction of the event handler. (In either case the micro-PC that is pushed onto the stack is the PC of the micro-instruction that was inhibited or stopped.)

MACRO EVENTS

The following is a list of macro events, divided into arbitrary classes.

	condition	mask	priority
MEMORY			
refresh memory	X		2
SEQUENCER			
IBUF_empty	X	X	3
resolve_links	X	X	6
CSA_underflow	X	X	4
CSA_overflow	X	X	5
EXTERNALS			
sysbus	X	X	
timer	X	X	
diagnostic intr.	X	X	
power failure	X	X	1

All macro-events will only occur during a dispatching instruction. The dispatch micro-instruction will complete entirely, except the following instruction will be the first micro-instruction of the macro event handler. Most of the macro events are testable conditions and most are maskable.

BRANCH CONTROL

The micro-sequencer allows 16 possible branch types which are as follows:

```
brt             --conditional branch (branch if true)
brf             --conditional branch (branch if false)
br              --unconditional branch
cont            --continue (PC + 1)
callt           --conditional call (call if true)
callf           --conditional call (call if false)
call            --unconditional call
returnt         --conditional return (return if true)
returnf         --conditional return (return if false)
return          --unconditional return
dispt           --conditional dispatch (dispatch if true)
dispf           --conditional dispatch (dispatch if false)
disp            --unconditional dispatch
case            --jump to the branch address plus the lsb 14 bits
                --of the FIU_DATA from the last cycle
case_call       --same as the case, except PC + 1 is pushed onto
                --the stack
push            --push the branch address onto the micro_stack
```

The next micro-address for each combination of condition value and branch type is shown in the following table.

branch_type	condition value true	condition value false
brt	branch_addr	PC + 1
brf	PC + 1	branch_addr
br	branch_addr	branch_addr
cont	PC + 1	PC + 1
callt	branch_addr	PC + 1
callf	PC + 1	branch_addr
call	branch_addr	branch_addr
returnt	micro_stack	PC + 1
returnf	PC + 1	micro_stack
return	micro_stack	micro_stack
dispt	decode_ram	PC + 1
dispf	PC + 1	decode_ram
disp	decode_ram	decode_ram
case	branch_addr + FIU_data(0:14)	--for both cases
case_call	"	" -- "
push	PC + 1	PC + 1

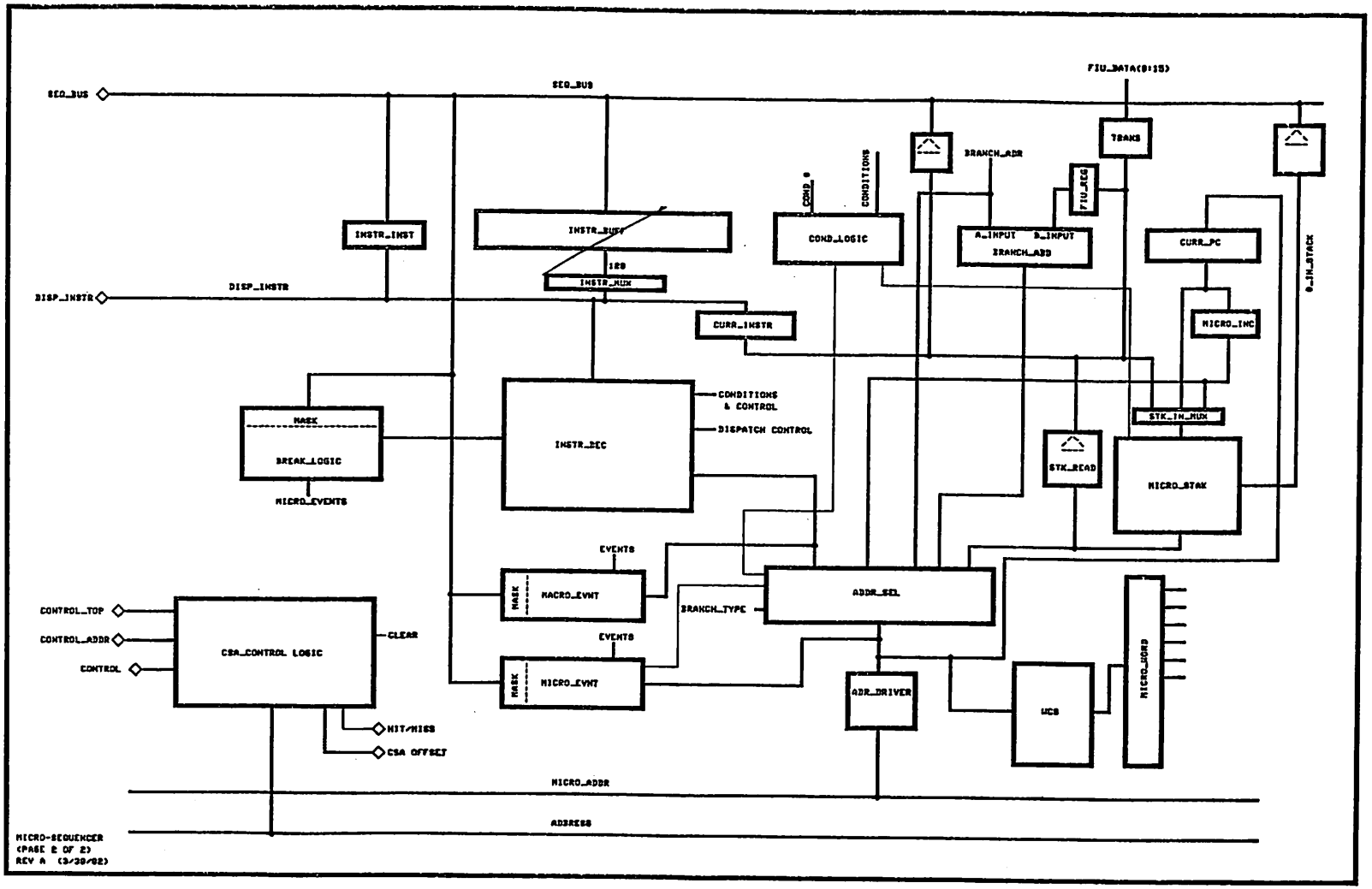
BRANCH_ADDR

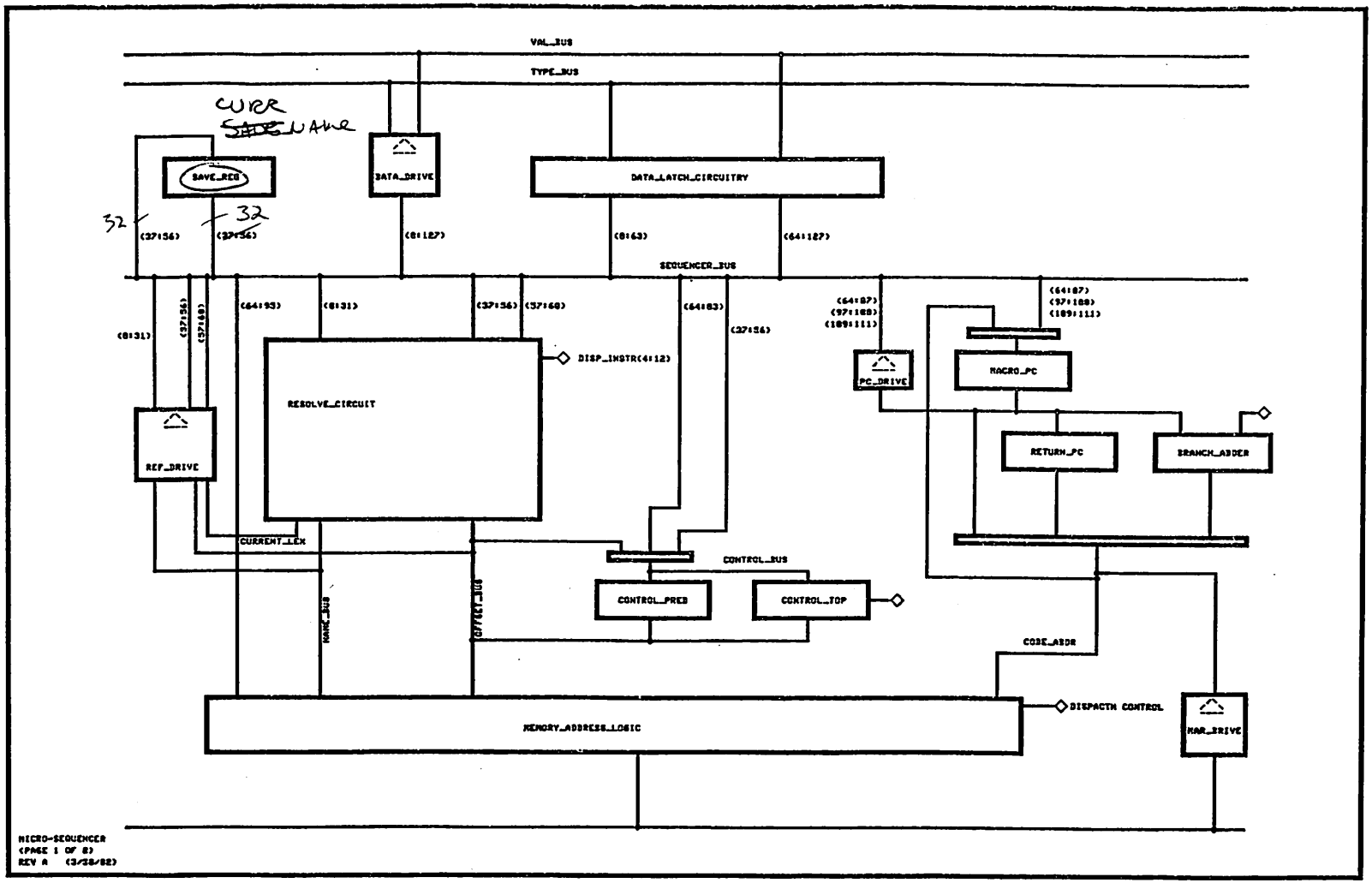
Each hardware board has a micro-field that selects one of the testable conditions on that board (this condition may be a combination of conditions on that board). The micro-sequencer has a micro-field that selects which of these board conditions are tested during this micro-instruction (this tested condition may be a combination of the board conditions, or a combination of the board conditions and the latched condition).

The micro-sequencer contains a latch that will latch the state of the currently tested condition under microcode control. One bit of the micro word specifies if the currently seleted condition is to be latched. (This latched condition may be selected as a branch condition, see branch timing below.)

Because of the timing for the conditions some occur early in a micro-cycle and some occur late. Early conditions may always be selected for branches. Late conditions, if selected for a branch condition, must be followed by a hint. The hint informs the micro-sequencer that usually the branch will fail (take PC + 1) or usually the branch will succeed (take the branch address). If the hint is not correct the micro-sequencer will not execute the selected micro-instruction, but will take one micro-cycle to calculate the correct micro-address. The branch timing bits for the early/late/hint conditions are interpreted as follows.

branch_timing		
0	0	branch on an early condition
0	1	branch on a latched condition
1	0	branch on a late condition, hint is PC + 1
1	1	branch on a late condition, hint is take the branch


SEQ_BUS
SEQ_BUS
FIU_DATA(0:15)
INSTR_INST
INSTR_BUFF
128
INSTR_MUX
DISP_INSTR
DISP_INSTR
COND_LOGIC
COND #
CONDITIONS
BRANCH_ADR
A_INPUT
B_INPUT
BRANCH_ADD
FIU_REG
TRAPS
CURR_PC
MICRO_INC
B_IN_STACK
CURR_INSTR
MASK
BREAK_LOGIC
MICRO_EVENTS
INSTR_DEC
CONDITIONS & CONTROL
DISPATCH CONTROL
STK_IN_MUX
STK_READ
MICRO_STAK
EVENTS
MASK
MACRO_EVNT
BRANCH_TYPE
ADDR_SEL
CONTROL_TOP
CONTROL_ADDR
CONTROL
CSA_CONTROL LOGIC
CLEAR
EVENTS
MASK
MICRO_EVNT
ADR_DRIVER
UCS
MICRO_WORD
HIT/MISS
CSA OFFSET
MICRO_ADDR
ADDRESS
MICRO-SEQUENCER
(PAGE 2 OF 2)
REV A (3/30/82)


VAL_BUS
TYPE_BUS
CURR
SAVE_REG
DATA_DRIVE
DATA_LATCH_CIRCUITRY
32
32
(37:56)
(37:56)
(0:127)
(0:63)
(64:127)
SEQUENCER_BUS
(64:95)
(0:31)
(37:56)
(57:60)
(64:83)
(37:56)
(64:87)
(97:100)
(109:111)
(64:87)
(97:100)
(109:111)
(0:31)
(37:56)
(57:60)
DISP_INSTR(4:12)
RESOLVE_CIRCUIT
REF_DRIVE
PC_DRIVE
MACRO_PC
RETURN_PC
BRANCH_ADDER
CURRENT_LEX
CONTROL_BUS
CONTROL_PREB
CONTROL_TOP
NAME_BUS
OFFSET_BUS
CODE_ADDR
DISPACTH CONTROL
MEMORY_ADDRESS_LOGIC
MAR_DRIVE
MICRO-SEQUENCER
(PAGE 1 OF 2)
REV A (3/28/82)

```
MICRO EVENT CONTROL --1 bit
  disable all micro events  --1 bit

MACRO EVENT CONTROL  --2 bits
  disable all macro events  --1 bit
  disable dispatch event -- 1 bit  (during this cycle)

WRITE STROBES  --5 bits
  resolve circuit --1 bit
  macro_PC  --1 bit
  return_PC  --1 bit
  instr_buff --1 bit
  insert_instr  --1 bit

  COMMON WRITE SELECTS --4 bits
    break_mask
    micro_mask
    macro_mask
    clear CSA
    control_pred
    control_top
    save_reg
    cond(if false) load macro_PC & instr_buff
    push micro stack
    pop micro stack
    NOP

MEMORY ADDRESS CONTROL --4 bits
  memory type  --3 bits
   (data, code, control1, control2, type, queue, import, nop)
  addr from sequencer --1 bit
```

MICRO SEQUENCER MICROCODE BITS 3/30/8
(CURRENT TOTAL 47)

DATA LATCH CONTROL --2 bits
- open latch & validate
- open latch & invalidate
- invalidate ---not sure if this is necessary
- nop

DATA DRIVE CONTROL --2 bits
- nop, val, type, both

SEQUENCER BUS ENABLES high half (0:63) --2 bits
- latch
- name_bus, ll, offset_bus
- name_bus, ll, save_bus

SEQUENCER BUS ENABLES low half (64:127) --2 bits
- latch
- PC drive
- assorted masks (break, micro, and macro)

RESOLVE CONTROL --6 bits

VALID BITS --2 bits
- clear >ll
- set ll
- clear ll
- nop

LEX LEVEL CONTROL -- 2 bits
- current lex
- seq_bus (57:60) - 1
- 4 lsb of branch address

WRITE CONTROL --2 bits
- name, offset, and current_ll
- name, offset
- name
- offset

CONTROL MUX --2 bits
- offset_bus
- seq_bus (64:83)
- seq_bus (37:56)

OFFSET BUS --2 bits
- resolve output
- control_pred
- control_top

CODE ADDR MUX --1 bits
- return_PC
- MAR

CONDITION BITS --25 bits
- branch type --4 bits
- latch condition --1 bit
- branch timing --2 bits
- condition select --4 bits
- branch address --14 bits