CHAPTER 1 EUNITS INTRODUCTION

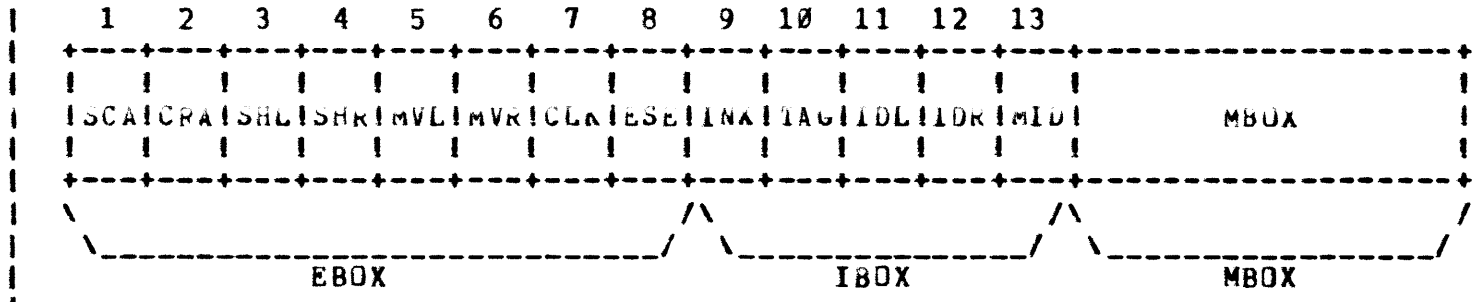
//TOC//

CHAPTER 1

EUNITS INTRODUCTION

1.1 EXECUTION UNITS PARTITIONING

JUPITER CPU PARTITIONING



JUPITER FPA PARTITIONING

1.1.1 Overall

The JUPITER CPU, in its maximum configuration, contains 3 units which are directly involved with the execution of instructions: IBOX, EBOX, and FPA. The IBOX fetches streams of instructions, performs effective address calculations, and fetches the memory operand, (or operands), necessary for the instructions it determines may be required by the EBOX. The EBOX accepts from the IBOX: AC addresses, an instruction code, a memory operand (or operands), and paging or error information pertaining to memory references of each instruction, as each is called up for execution by the "last cycle" micro-order. It then performs the operation indicated by the instruction code. The FPA provides a speedup of multiply, floating divide, floating add, and instructions requiring two memory fetches, and as such is a logical (and optional) extension of the IBOX and EBOX.

1.1.2 MID

MID contains FRU logic for the CPU (a callout of most probable failing module) , and IBOX-MBOX interface logic.

1.1.3 IDR

IDR (IBOX Datapath Right) contains bits 18:35 of the IBOX memory address path, operand data path and program counter logic. Instruction memory address conflict detection for VMA 27:35 and instruction loop detection (within the current PC section) are also located on this module.

1.1.4 IDL

IDL (IBOX Datapath Left) contains bits 6:17 of the IBOX memory address path and program counter logic, bits 0:17 of the operand data path logic, instruction code and AC address buffers, skip/jump determination logic, and ISET first cycle execution control RAMs.

1.1.5 TAG

TAG contains control logic for IDR and IDL. The TAG is a 4 bit value representing the PC address of an instruction which has been assigned to the IBOX by IPUT (IBOX Micro-Code). The TAG is used to address 16 word register files into which will be placed information pertinent to the instruction, such as instruction code, global bit, AC address, effective address calculation, memory operand, jump target address, etc. IPUT accepts priority trap requests from EBOX control and IBOX state logic, and thereupon directs address and operand path flow and TAG assignments.

1.1.6 INX

INX contains the write compare indexes for instruction bits 9:12(XR AC), instruction bits 14:17(AC), and OP2 address bits 27:35. These compares are used to initiate refetch operations for buffer positions within the IBOX that have been possibly altered by the EBOX. Also located on INX are the valid code logic, OP2 AC address update, and the clock start-stop mechanism.

1.1.7 MVR

MVR contains bits 9:17 and 27:35 of the fast execution path of the EBOX. It executes binary adds and subtracts, decimal adds and subtracts, Boolean functions, halfword operations, and mask and test operations. A duplicate copy of all AC sets and temporary registers <refered to as the master AC backup> is located on this module.

1.1.8 MVL

MVL is physically identical to MVR, and performs the same function as MVR with logical bits 0:8 and 18:26 of the fast execution path of the EBOX.

1.1.9 ESE

ESE contains EBOX microcode RAMS which control instruction execution during the first (1st 22ns period after which operands become available) and subsequent cycles. First cycle control comes from fast 256X4 RAMS, which setup EBOX control fields directly from lookup tables addressed by the instruction code. Second cycle control comes from slow 4KX1 RAMS which are also addressed by the instruction code. Third and subsequent cycle control comes from the same slow 4KX1 RAMS addressed by the next address path.

1.1.10 CRA

CRA contains slow 4KX1 RAMS which control sections of the EBOX during the 2nd and subsequent cycles of execution. During the first cycle some control fields default to specific options which allow data path gating via forwarding controls. Next address control and the trap and interrupt mechanisms are also located on this module.

1.1.11 SHR

SHR contains bits 18:35 of the shifter (which performs a 72 input, 36 output left shift), bits 18:35 of two REGFILES and two copies of the current AC set (one normally addressed by bits 32:35 of the effective address calculation, the other normally addressed by the AC field of the instruction), and bits 18:35 of the master AC set, a RAM file containing 8 AC sets plus 128 temporary registers.

1.1.12 SHL

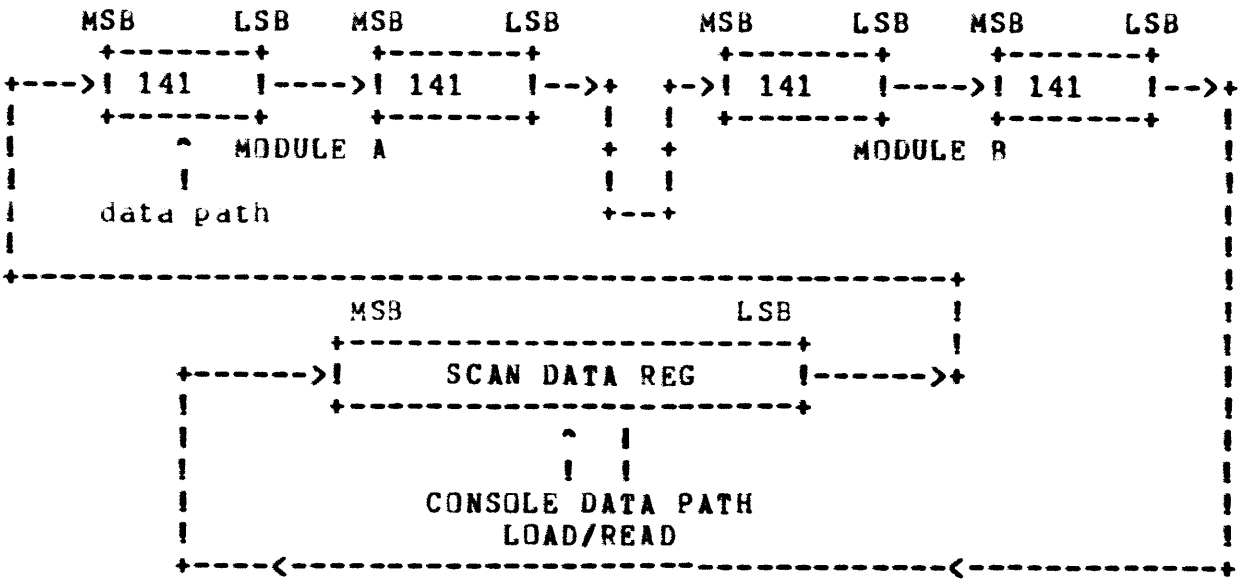
SHL contains bits Ø:17 of the shifter, register files, AC copies, and master AC set.

1.1.13 SCA

SCA contains a 13 bit shift control path with addition and subtraction capability. Also located on this module are an accounting meter, interval timer, 1US time base, PC flags associated with shift and floating point operations, the state reg, and APR flags.

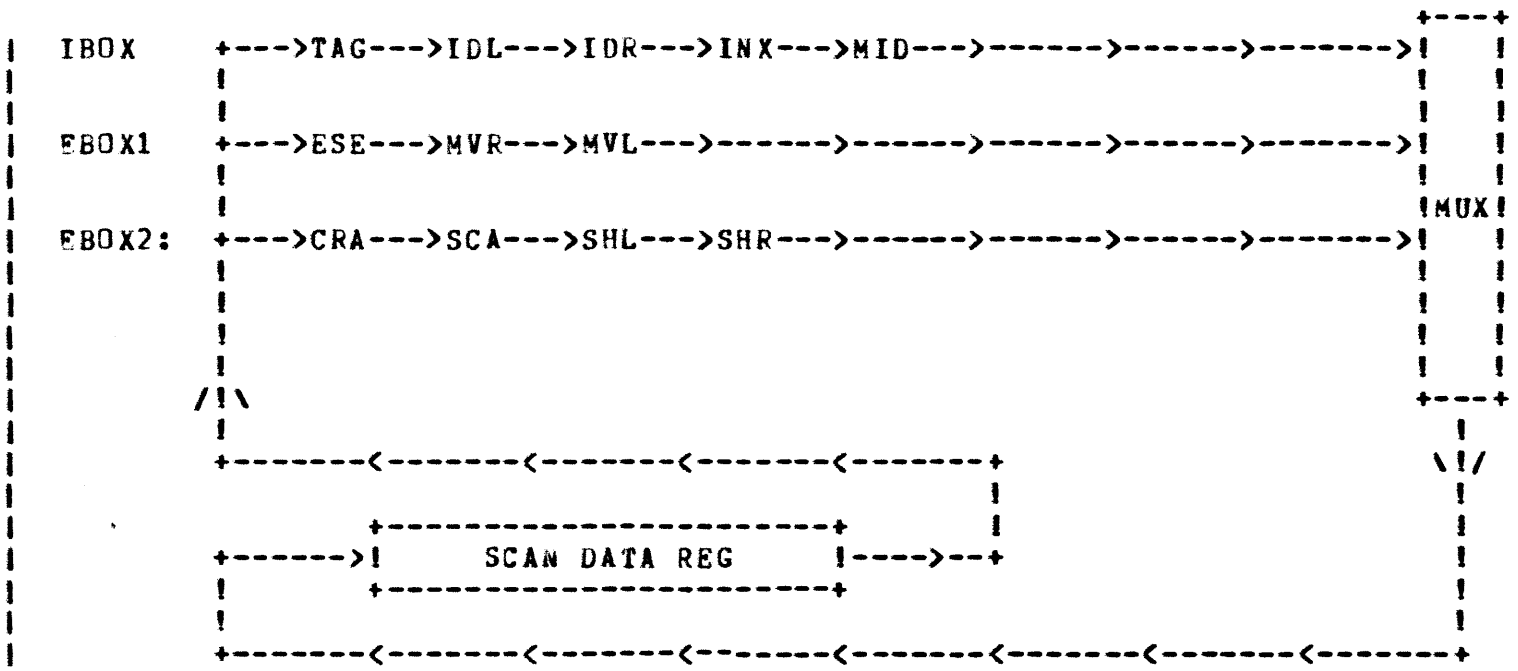
1.2 DIAGNOSTIC INTERFACE

The diagnostic interface is the vehicle through which the console A) presets execution unit registers to desired states, B) forces control decodes to manipulate the data paths, C) gains visibility into the data and control paths, and D) loads control storage areas. The 100141 shift register is used to achieve all of these. It provides the ability to locate the scan in/scan out mechanism as a functional part of the data path. All 100141 shift registers used in this manner and strung together (one's shift output connected to the other's shift input) are considered to be part of the active scan path. Other 100141 shift registers, not part of functional logic, are used for presetting signals to desired states, and scanning data path and control states independently of E unit operation. These are linked together as the passive data path.



1.2.1 Active Scan Paths

The JUPITER CPU contains five active scan paths. Each is sourced at the IOBOX and consists of shift registers which, under normal operation (not clocks stopped due to error or console control) perform a parallel load when their corresponding system clocks occur. When the system clocks have been blocked by error freeze or a clock control stop, the console, via the IOBOX may perform one of two operations; A) a parallel load via its own clock, B) a shift (normally hi-order to lo-order) of 1. The active scan paths are connected as follows (MBOX path shown in MBOX spec):



1.2.2 Passive Scan Path

The JUPITER CPU contains two passive scan loops, one specifically for the CLK module, another for the IBOX, EBOX, and MBOX modules. They may be accessed at any time, without stopping the system clocks. A strobe,synchronized to system clocks, is used to gate controls to perform various console-initiated operations. The passive scan path is used for the following purposes:

1. Real-time scan of PC, VMA, FRU CODE
2. Force errors, force L MUX clock, disable error stop, enable auto retry
3. Strobed write to microcode RAMs and lookup tables, error reset, clear data paths, clock restart, reset FRU stored

1.2.3 Scan Read And Load

A 16 bit shift scan data register in the IOBOX completes the scan path loop. The register is loaded from the console via a 16 bit parallel load path. The LSB of the register fans to the inputs of all active and passive scan paths. The MSB shift in signal comes from an 8-way mux of scan path LSBs.A common shift line is used for EBOX1, EBOX2, IBOX, and FPA. Each scan loop has a separate DIAG CLOCK. The Scan Data Register represents a 16 bit window within the selected scan loop. To read a 16 bit segment of the selected loop, it is shifted the appropriate number of times needed to position that segment in the Scan Data Register. It may then be read and/or altered directly by the console. The segment is then shifted back to its original position in the scan loop.

1.2.4 Reset

Reset of EBOX and IBOX is achieved by two methods: 1) Turning on SPFN REG 3 by the EBOX microcode will cause a simultaneous clear of all EBOX and IBOX data paths and error latches, with the exception of ESE and CRA data paths. 2) Diagnostic load of passive scan register bits at the CLK module followed by a diagnostic strobe will permit individual control of the following:

a) clear EBOX error latches
b) clear IBOX error latches
c) clear ESE data paths
d) clear EBOX, IBOX data paths (excluding ESE and CRA)
e) clear MBOX error latches
f) clear MBOX data paths
g) clear current AC sets at EBOX and IBOX

3) Diagnostic load of a passive scan register bit at CRA will permit clearing of the CRA module data paths.

1.3 ERROR CONTROL LOGIC

Error control logic for JUPITER is designed in such a way as to support 3 RAMP goals; a) detection and recording of intermittant as well as solid failures, b) hardware recovery from a high percentage of CPU errors, c) fault isolation <to a module> of a high percentage of errors prior to running of diagnostics and, in the case of intermittants, without taking the system down. 20 to 25 % of JUPITER logic is devoted to the implementation of these features.

1.3.1 Error Detection And Recording

Odd parity is generated for every bus greater than 4 bits wide. A high emphasis is placed in three areas:

1. Fault isolation by module
2. Console access to internal data paths
3. "Catching" intermittant errors as close as possible to the point of failure.

36 bit data paths develop 1 parity bit for every 9 data bits. Microcode storage areas contain 1 parity bit per word except in the case of EBOX control storage, which contains 4 parity bits, two used for module fault isolation, and the other two for number field parity. The EBOX main ALU, EBOX SCA ALU, and IBOX EA ALU are compared to a duplicate every cycle. Parity generation occurs at the output of boolean, ALU, and shift functions; parity checking occurs at the input of boolean, ALU, and shift functions. Parity is modified and propagated with 1 and 2 bit shifts. A modulo 3 residue checking system is employed for verification of all FPA multiply and divide results.

The scan-in, scan-out mechanism described above is the method by which console may assert control or data path signals, and read out a high percentage of the data path. 100141 shift registers constitute approximately 12% of JUPITER execution unit logic. Diagnostic control has reserved to it a section of the microcode control storage, thus allowing it to route, via set up of next address to read a microword with desired control decodes, virtually all registers (if not already a 141) through mux paths and into 141s for scanning. This includes access to 16 word register files and RAM arrays.

Strategic registers are frozen if an error is detected at their output. This approach helps in locating the source of an error likely to be propagated through the system or lost, by the time the IBOX, EBOX, and FPA module clocks can be stopped.

1.3.2 HARDWARE RECOVERY

When an error has been detected, an error latch is set at the module level. The error condition may then block the next clock to the failing register. The error signal is also sent to the INX module where it is funneled into the clock holdoff latch, which will then stop the next clock Ø, 1 , 2, and 3 in succession. The error condition will block the store of results to ACs or memory for all cycles following initial error detection, until the error latch is reset by it's designated CLEAR ERRS signal. A 2 bit code, called the Retry Code, is loaded with the FRU reg to identify the number of instructions which have entered the pipe since the initiation of the instruction causing the error. The code is used by the console to determine the number of PC instructions to step back in preparation for a retry of the failing instruction . The console may also elect to arm the EBOX AUTO RETRY latch, which will force a microcode trap on an error stop condition. A retry routine will branch on the value of the retry code to a microword containing the appropriate command to the IBOX. The Retry Code is defined as follows:

RETRY CODE	RECOVERY
Ø	restart from PC BUFFER addressed by SEL I-1
1	restart from PC BUFFER addressed by SEL I-2
2	restart from PC BUFFER addressed by SEL I-2
3	restart from PC BUFFER addressed by SEL I-3

Successful retrys appear to be possible for nearly all instructions with only 1 result to be stored. However in instructions with multi-word results which overlay previous operands, successful retry is not probable.

1.3.3 Ebox Microcode Retry

1. TRAP to address 4Ø1Ø
2. Disable Traps
3. Dispatch on retry code and step back SELI history counter
4. Retrieve SELI and PC flags of instruction to be retried
   a. Save PC flags in MAC scratch pad location
5. Clear EBOX, IBOX, and FPA data paths
6. Read PC of new SELI into data path
7. Flush Pipe
8. Go to LOAD PC routine at IBOX
9. Enable Traps
10. Issue Last Cycle

1.3.4 Fault Isolation

For each error stop, the FRU REG is loaded at the MID module, provided that the FRU STORED F/F is off. The FRU REG identifies to the console the most probable failing module. The errors are prioritized so that the source of the error will be identified, rather than the path which

merely propagated or received erroneous data. Since the errors are frozen dynamically, and fed through a priority encode to generate the FRU CODE, an accurate method of trouble-shooting one error at a time is achieved. The FRU REG, in the passive scan path, may be scanned while the IBOX and EBOX clocks are running following an error detection and retry startup. The individual error conditions are also frozen upon detection and are scannable via the active scan path prior to a console restart of system clocks. During the microcode recovery routine, a dispatch on pipe error occurs, causing the EBOX to initiate a flush of the IBOX pipeline.

Reloadable storage areas, shown below, are handled by the console as a special case.

IPUT microcode	TAG module
ED table lookup	TAG module
ISET microcode	IDL module
ESE microcode	ESE module
CRA microcode	CRA module
SHL MAC	SHL module
SHR MAC	SHR module
FPA microcode	ARC module
Divide Lookup	OPF module

A failure at reloadable storage will block Auto Retry, even if console has enabled the Auto Retry mechanism. Console, after reloading the failing area, must force the EBOX microcode to enter its retry routine via the CRA load address reg. Each of the above areas has a backup. In the case of a microcode or divide lookup error, the diagnostic load address reg for that storage contains the failing address; therefore console may reload that word with data from it's own microstorage backup. In the case of a MAC error, the hi order 4 bits of the write address are held at the MAC backup on MVL and MVR, and will be used by the console to reload the failing 16 word set from MAC backup.

.BB

Below is a list of errors versus FRU codes in order of priority.

ERROR	FRU CODE
IBOX HI ERRORS:	
IPUT MICROCODE ERROR	Ø TAG
ED TABLE LOOKUP ERROR	1 TAG
PF OR TAG IN ERROR	2 MID
LEFT E1E2 ERROR	3 IDL
RIGHT E1E2 ERROR	4 IDR
IC ERROR	5 IDL
STR ERROR	6 TAG
INX ERROR	7 INX
IBOX LO ERRORS:	
ISET ERROR	Ø IDL
LEFT EA ALU ERROR	1 IDL
RIGHT EA ALU ERROR	2 IDR
MVL MAC BACKUP ERROR	3 MVL
MVR MAC BACKUP ERROR	4 MVR
LEFT OP2 ERROR	5 IDL
RIGHT OP2 ERROR	6 IDR
VCODE OR VBUF ERROR	7 INX
EBOX HI ERRORS:	
LEFT MAC ERROR	Ø SHL
RIGHT MAC ERROR	1 SHR
FLAGS ERROR	2 SCA
RIGHT X,Y ERROR	3 SHR
SCA INTERNAL ERROR	4 SCA
ILLEGAL PAGE FAULT	5
LEFT X,Y ERROR	6 SHL
SPARE	7
EBOX LO ERRORS:	
ESE ERROR	Ø ESE
CRA ERROR	1 CRA
MVL OUTPUT ERROR	2 MVL
MVR OUTPUT ERROR	3 MVR
MVL ALU ERROR	4 MVL
MVR ALU ERROR	5 MVR
MVL INPUT ERROR	6 MVL
MVR INPUT ERROR	7 MVR

FPA ERRORS
<TO BE DEFINED>

1.4 CLK MODULE SPECIFICATION

The JUPITER clock controls provide the gating of a master oscillator source in such a way that synchronous clock phases will result at the IOBOX, IBOX, EBOX, FPA, MBOX, and memory array. The clock control logic will also include start, stop, and count mechanisms for diagnostic purposes. The CLK module also contains diagnostic logic which controls data path resets, error latch resets, and scan path operation within the CPU.

1.4.1 IOBOX INTERFACE TO THE SYSTEM CLOCK

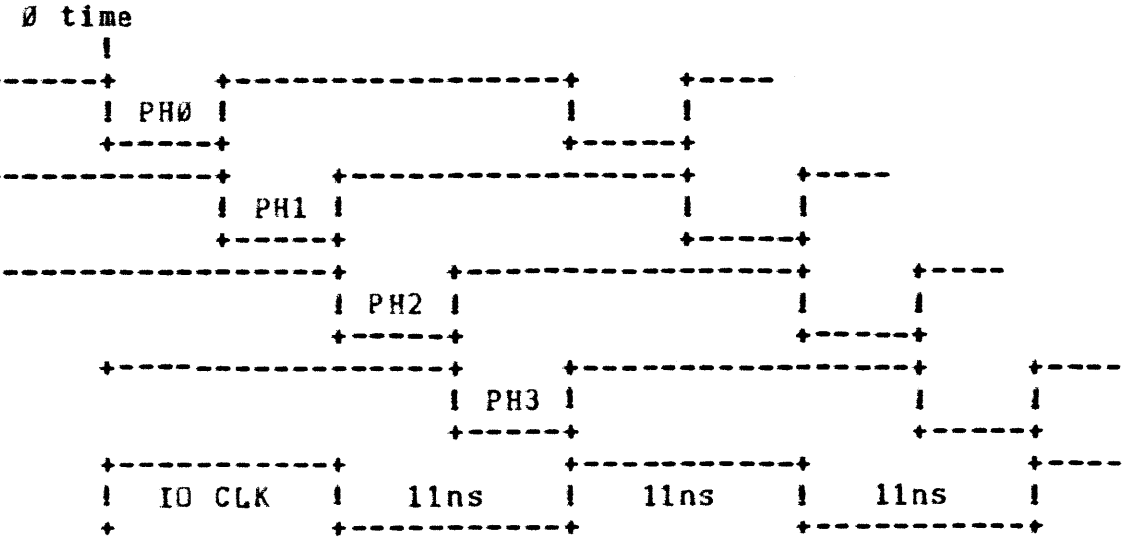
Outputs from CLK module to IOBOX:

1.4.1.1 FREE RUN CLK (source CLK2) -

The CLK module will send a 45.45 Mhz pulse, 11ns on, 11ns off, to the IOBOX. The positive transition of this pulse will correspond to the negative transition of phase Ø at the CPU. It will be free-running at all times. (dest IOPI)

1.4.1.2 IO CON CLK A,B (source CLK2) -

The CLK module will send two 45.45 Mhz pulses, 11ns on, 11ns off, to the IOBOX. The positive transition of these pulses will correspond to the negative transition of phase Ø at the CPU. It will be blocked by DISABLE M. (dest IOMG)



1.4.1.3 CLK ERR (source CLK6) -

CLK Error will be set if two or more phases of the four-phase clock ring become active simultaneously, or if a phase fails to occur as shown in the above dwg. It is cleared by RESET TO CLK MOD from the IOBOX. (dest IOPG, xlated to TTL and sent to CSLG)

1.4.1.4 CLK MOD SCAN IN (source CLK3) -

Output of the passive scan path at the CLK module. (dest IOPG)

1.4.2 Inputs To CLK Module From IOBOX:

The IOBOX sends controls to the CLK module to set up and execute the clock start/stop features. The passive scan path permits the IOBOX to shift into a control register a field representing a command to the clock control logic.

1.4.2.1 CLK MOD SCAN CLK (source IOPG) -

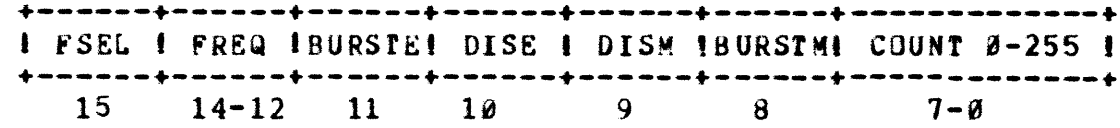
This clock will be used to load or shift the the clock control passive scan path. (dest CLK3)

1.4.2.2 CLK MOD SCAN OUT (dest CLK3) -

This is the serial data input to the clock control passive scan path. It will be used to load the Clock Control Register. (source IOPG)

Command Bits of the Clock Control Register:

CLOCK CONTROL REG



1.4.2.2.1 COUNT Field (bits 7-Ø) -

The COUNT field represents a count of clock phases to be stepped in the MBOX, IBOX, EBOX and FPA, when BURSTE or BURSTM are turned on. For example, Disable E (bit 1Ø) on will stop all clocks in the CPU following the next active phase 3 (after the positive transition of

phase 3, all EBOX clocks will be degated). Following this, a count of ØØ1 and BURSTE on will step all clocks in the CPU through the negative, then positive transition of phase Ø. Count increments of 4 represent full machine cycles (22ns).

1.4.2.2.2 BURSTM (bit 8) -

The BURSTM command will will cause the MBOX clocks to step the number of phases indicated by the COUNT field. The BURSTM control register bit will be reset when the count is completed.

1.4.2.2.3 DISABLE M (bit 9) -

The DISABLE M command will stop all MBOX clocks following the positive transition of the next phase 3. Turning DISABLE M off will start the MBOX clocks with the negative transition of the phase which would normally follow the last phase issued. The last phase issued is called PHASE X. Its value (Ø,1,2,or 3) is latched when a burst count completes, when DISABLE E or DISABLE M are turned on, or when it is altered via console load from the CLK module passive scan path.

1.4.2.2.4 DISABLE E (bit 1Ø) -

The DISABLE E command will stop all EBOX, FPA and EBOX controlled IBOX clocks following the positive transition of the next phase 3. Turning DISABLE E off will start the EBOX, FPA, and EBOX controlled IBOX clocks with the negative transition of the phase following PHASE X (see DISABLE M for a description of PHASE X).

1.4.2.2.5 BURSTE (bit 11) -

The BURSTE command will step all EBOX, FPA, and EBOX controlled IBOX clocks the number of phases indicated by the COUNT field. The BURSTE control register bit will be reset when the count is completed.

1.4.2.2.6 FREQ (bits 12-14) -

The machine cycle frequency may be varied from the nominal 22ns by the FREQUENCY CONTROL bits 12-14. The selections are:

Ø	EXT source
1	2ØØ Mhz, 2Øns machine cycle
2	181.8 Mhz (nominal), 22ns machine cycle
3	163.6 Mhz, 24ns machine cycle
4	nominal div by 2, 44ns machine cycle

FREQ SEL ENA (bit 15)

The FREQUENCY SELECT ENABLE will cause the frequency of the clock to change according to the FREQUENCY CONTROL bits.

1.4.2.3 CLK MOD SET ECL (source IOPG) -

This the strobe which will load the Clock Control Register from the clock control passive scan path. (dest CLK3)

1.4.2.4 CLK PAS TO CLK (source IOP?) -

Clock to the diagnostic passive scan path. (dest CLKB)

1.4.2.5 CPU ACT SCAN (source IOPI) -

Controls whether the active scan paths in the CPU are in serial shift (CPU ACT SCAN on) or parallel load mode (CPU ACT SCAN off). (dest CLKB)

1.4.2.6 CPU SCAN OUT ECL (source IOPG) -

The serial data input to the active scan paths of the CPU. (dest CLKB)

1.4.2.7 CPU STROBE (source IOPI) -

The strobe which will enable the control information of the diagnostic passive scan path in the CPU. (dest CLKB)

1.4.2.8 PAS SCAN MODE (source IOPI) -

Controls whether the diagnostic passive scan path in the CPU is in serial shift (PAS SCAN MODE on) or parallel load mode (PAS SCAN MODE off). (dest CLKB)

1.4.2.9 RESET TO CLK MOD (source IOPG) -

Clears CLK ERR and restarts the clock source ring. (dest CLK3)

1.4.2.10 CLK SCAN MODE (source IOPG) -

Controls whether the clock control passive scan path is in serial shift (CLK SCAN MODE on) or parallel load mode (CLK SCAN MODE off). (dest CLK3)

1.4.3 CPU INTERFACE TO THE SYSTEM CLOCK

Outputs From CLK Module To The CPU:
PHASE 0 TO (module name)
PHASE 1 TO " "
PHASE 2 TO " "
PHASE 3 TO " "
Two identical sets will go to the IBOX modules to allow for separate control by DISABLE M and DISABLE E.

1.4.3.1 DIAG CMD ENABLE (source CLKG) -

Console-initiated strobe to the passive scan controls at CRA. (dest CRAA)

1.4.3.2 DIAG SH ACTIVE EBOX (source CLKG) -

When this signal is high, the scan path is in serial shift mode; when low, the scan path is in parallel load mode. (dest SCAH, CRAH, SHLM, SHRM, MVLF, MVRF, ESEI)

1.4.3.3 EBOX1 ACT DATA IN (source CLKG) -

Serial data input from the console to the EBOX1 active scan path. (dest ESFG)

1.4.3.4 EBOX2 ACT DATA IN (source CLKG) -

Serial data input from the console to the EBOX2 active scan path. (dest CRAI)

1.4.3.5 PRESET AC LEFT, RIGHT (source CLKG) -

Console controlled clear of the register files which contain the current AC set. Used for EBOX initialization.(dest SHR8, SHL8)

1.4.3.6 RESET CRA ERROR (source CLKG) -

Console controlled clear of error latches on CRA and SCA modules. (dest SCA2, CRA7)

1.4.3.7 RESET SCA (source CLKG) -

Console controlled clear of flags, latches, and data paths on SCA. (dest SCA2)

1.4.3.8 TOD CLK (source CLK1) -

TOD CLK is a 50Mhz, 20ns on, 20ns off signal sent to the MID module which will be further divided to generated the step to the time base and interval counter at the SCA module. (dest MIDB)

1.4.4 Inputs To CLK Module From The CPU:

1.4.4.1 CLK HLD OFF F+E (dest CLK7) -

CLK HLD OFF F+E from the CPU INX module will stop all clocks to the EBOX and FPA following the positive transition of the next phase 3. CLK HLD OFF F+E going inactive will cause the same clocks to start up with the negative transition of phase 0. (source INXC)

1.4.4.2 CLK HLD OFF IBOX (dest CLK7) -

CLK HLF OFF IBOX from the CPU INX module will stop all EBOX related IBOX clocks following the positive transition of the next phase 3. CLK HLD OFF IBOX going inactive will cause the same clocks to start up with the negative transition of phase 0. (source INXC)

1.4.4.3 DIAG PAS OUT MID (source MID2) -

Serial data in to the clock module portion of the diagnostic passive scan path. (dest CLKB)

1.4.4.4 SPFN CLR DP ERR (source SCAH) -

Resulting from EBOX LD SPFN micro-order and # Field Ø3, causes reset of EBOX and IBOX data paths and error latches with the exception of CRA and ESE data paths. The CLK module is used to drive the resulting control signals to all modules affected. (dest CLKB)

1.4.5 MEMORY INTERFACE TO THE SYSTEM CLOCK

Outputs From Clock Control To Memory Modules:

1.4.5.1 EARLY PHØ TO MMC (source CLK2) -

An individually driven, free-running phase Ø will be sent to the MMC module in the MBOX. From there it will be re-powered with copies sent to the memory array modules. (dest MMCR)

CHAPTER 2

EBOX

2.1 GENERAL

The JUPITER EBOX is designed in such a way that maximum performance is derived from a limited number of ICs. Emphasis is placed on the fast execution of commonly used instruction groups including mask and test, full and halfword moves, arithmetic, jumps, skips, and boolean. In approaching the design, it was found to be more efficient to make an entire group run fast rather then focus on a few highly used instructions. This in turn also served to reduce considerably the amount of control logic required to deal with the exception cases.

Since instruction mix analysis showed that the above-named groups occupied from 80-90% of dynamic execution, it was decided to create a hierarchical design, with MOVE <executing move and mask&test> and ALU <executing arithmetic, stack, and boolean> controlled from fast 256X4 RAMS, and all other instructions executed in the traditional method of microprogram control. Thus, the JUPITER EBOX executes nearly 300 instrucions in one or two 22ns cycles.

22ns prior to the start of execution, the IBOX supplies the EBOX with the instruction code, AC addresses <for both AC field and EACALC 32:35>, and an instruction/operand valid bit which permits the instruction execution to proceed. The instruction code addresses the fast 256X4 and slower 4kX1 RAMs simultaneously. In the next cycle <first cycle of execution>, the microword controlling the EBOX is selected from the fast RAMs. The second cycle of execution may be either a 22ns cycle or a 44ns cycle depending on the setting of the LAST CYCLE and L EARLY bits < LAST CYCLE and L EARLY both on indicate a 22ns cycle>. The second and subsequent microwords words are selected from the 4kX1 RAMs each 44ns.

The fast 256X4 RAMS control only the 22ns path <MOVE, ALU>, whereas the 4kX1 RAMs control a 44ns shift and shift control path as well as utilizing the MOVE and ALU sections as part of a larger 44ns path. This provides the microprogram with a very powerful arithmetic, boolean, merge, compare, and shift capability.

2.2 MVE MODULE (MVL,MVR IN CPU BACKPANEL SLOTS 5,6)

2.2.1 MVE Functionality

2.2.1.1 MOVE/ALU Path - The MOVE/ALU path is contained on two identical modules, MVL and MVR. It consists of two 22ns loops, each of which may receive inputs from the other, or from SHIFT , FPA, or IBOX operand paths, under the control of X1, Y1, or RF fields.

2.2.1.2 MOVE Operation - Two operands are presented to the MOVE path at CLK 2, SEL OP1 and SEL OP2. SEL OP1 in the first cycle of execution receives a word from one of the following:

1. X or Y AC sets < X is addressed by EACALC 32:35, and Y is addressed by AC field> on the X or Y BUS
2. A forwarded result through L1, D, and X or Y <if current instruction requires an operand from an AC word which is in the process of being stored
3. F BUS < if conflict compare indicates that previous instruction result selected at L MUX is from F and current instruction uses it as an operand>
4. RF BUS < if conflict compare indicates that previous instruction result selected at L MUX is from RF and current instruction uses it as an operand>. SEL OP2 in the first cycle of execution receives a word from one of the following:
   a)the X AC on the X BUS <X AC addressed by EACALC 32:35>
   b)a forwarded result through L1, D, and X <if current instruction requires an operand from an AC word which is in the process of being stored>
5. The IBOX OP2 BUS if no results are being forwarded and the EACALC does not represent an AC
6. From R BUS or F BUS via the same forwarding controls applied to SEL OP1.

In second and subsequent cycles SEL OP1 and SEL OP2 inputs are controlled entirely from 4kX1 control storage RAM fields X1, Y1, and RF.

Half word instructions are executed via SW and R MUX controls. SW MUX permits merging of either left or right half of SEL OP2 with the left or right half of SEL OP1. R MUX may select either half= zeros or ones, or either half = SEL OP2 bit 0 or 18. Move, Zeros, and Ones instructions are also executed via SW and R MUX controls.

Masking for test instructions occurs at an AND function with inputs from SEL OP1 and SEL OP2. The AND true outputs connect to comparators which, under COMP field control perform a partial compare <in the first cycle> of the AND outputs to the values -1, 0, and +1. Compare

results go to the IBOX where skip/ jump successful determination will be made. In the case of TEST LEFT instructions, SEL OP1 receives the AC operand with it's left and right halves swapped at the OP1 MUX, and SEL OP2 receives OP2 BUS from the IBOX with zeros in it's left half, and the lo order 18 bits of the EACALC in the right half.

All MOVE path results are clocked into the RF MUX at a delayed CLK 3 of the same cycle, <CLK 3 gated by L EARLY B from the current cycle's microword>.

2.2.1.3 ALU Operation -

ALU inputs are logically identical to MVE inputs, with the exception that in the first cycle, the Y1 MUX may be disabled to force zeros into the Y1 ALU input. This is used with an ALU decode 4 and forced carry to provide a fast add 1 or subtract 1 to either ALU half, while AF MUX, also under Y1 FIELD control, is gating the AC operand into the MOVE path.

The main ALU is 36 bits wide, divided into two 18 bit sections. an extension of two bits is made on the high order end to facilitate multiply operations, thus providing F BUS -2 and -1 to the SHL module. The ALU performs binary and decimal adds and subtracts, and boolean functions in 22ns. The ALU 0:35 may also be separated into two 18 bit sections with carry control to both.

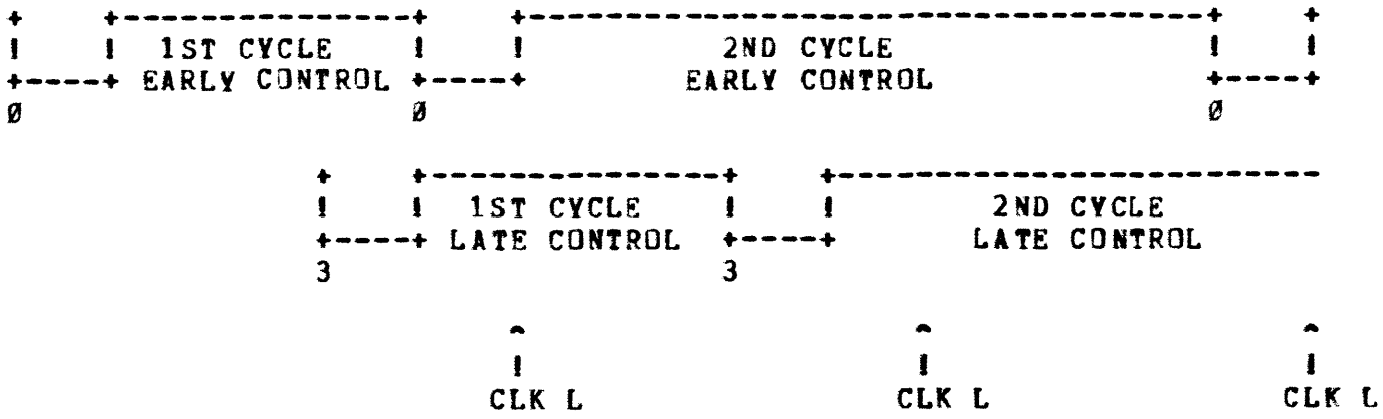
ALU results are clocked into the F MUX at CLK 2 of the same cycle <CLK 2 gated by L EARLY B of the current cycle's microword>.

2.2.1.4 L BUS - The L BUS is used to transfer results from the R BUS <MOVE>, F BUS <ALU>, or A BUS <FPA> to the following modules:

negative L BUS <1 backpanel stub at IDL,IDR>
- a) IBOX IDL and IDR
- b) MBOX MDL and MDR

positive L BUS <2 backpanel stubs, one each at SHF and FPA>
- c) EBOX SHL and SHR
- d) FPA

The L MUX is clocked at CLK 1. Since it is a stubbed bus, 10ns are allowed before it's destination registers are clocked at CLK 3.



2.2.2 Error Control

Parity generation occurs at the output of the F BUS < ALU result> and at the output of the AND function of the MOVE path. A parity bit is generated for each byte, composed of bits Ø:8, 9:17, 18:26, and 27:35, respectively. Parity checking is performed at the output of the SEL OP1 MUX, SEL OP2 MUX, and L MUX. If a parity error is detected at either SEL OP1 or SEL OP2, the following CLK Ø will set the MVL or MVR INPUT ERROR latch which will a) block the next CLK 2 to SEL OP1, SEL OP2, and ALU, and b) will signal to the error control logic on the MID module to stop all gated IBOX, EBOX, and FPA clocks starting with CLK Ø 20ns later, and sequentially stopping CLK 1, CLK 2, and CLK 3, in that order. If a parity error is detected at the L MUX, the following CLK 3 will set the MVL or MVR OUTPUT ERROR latch which will a) block the next CLK 1 to the L MUX and leading edge control CLK 1 of the MAC BACKUP write-enables, and b) will signal the error control logic on the INX module to stop all gated IBOX, EBOX, and FPA clocks starting with CLK Ø 27.5ns later.

The carry generates of the two duplicate ALUs are compared to determine whether an ALU error has occurred. If the results of the two ALUs combined produce an odd count, an error trigger is set at the next CLK 3, followed by a clocking of the MVL or MVR ALU ERR latch at the following CLK Ø. The MVL or MVR ALU ERR signal will cause the INX error control to stop all IBOX, EBOX, and FPA clocks starting with CLK Ø 20ns later.

System reset will reset F, RF, and L MUXes to zeros with good parity, and OP1 and OP2 to ones with good parity. with good parity. Error reset will reset MV OUTPUT ERR, MV INPUT ERR, AND MV ALU ERR.

2.2.3 Interface: Inputs To MVL,MVR Modules

2.2.3.1 A BUS ØØ:35, 4P (source FPA) -

36 bit result word from FPA. (dest MVEG)

2.2.3.2 ALU Ø:3 UW ØØ:Ø3 (source ESEB, ESEC) -

Timing: CLK Ø gated by previous machine cycle = ODD CYCLE. ALU control field from EBOX microcode. See EBOX MICROCODE CONTROL SPECIFICATION under ALU. (dest MVEL, MVEM)

2.2.3.3 CC Ø:2 UW Ø4:Ø6 (source ESEB,C) -

Timing: CLK Ø gated by previous machine cycle = ODD CYCLE. Carry Control field from EBOX microcode, determining the mode of operation of carry into bit 17 and carry into bit 35 of the main ALU. See EBOX MICROCODE CONTROL SPECIFICATION under CARRY CONTROL. (dest MVEI)

2.2.3.4 CLEAR ERRS (source CLKB) -

Clears all MVE error latches. Set on by strobe from the console. (dest MVEK, MVEL)

2.2.3.5 CMPR TO 7S UW 17, CMPR TO 1 UW 18 (source ESED) -

Timing: CLK 1 gated by previous machine cycle = ODD CYCLE. Comp field from EBOX microcode, used for compare to zero, minus one, or plus one, of the result of a masking operation of SEL OP1 and SEL OP2. See EBOX MICROCODE CONTROL SPECIFICATION under COMP. (dest MVEM)

2.2.3.6 CRY To 35 (source MVLI) -

From carry control logic on MVLI, is used for carry lookahead there and is the sent to MVRI for the same purpose. (dest MVRI)i

2.2.3.7 DCAR Ø8, 17, 35 (source MVEI) -

Decimal add/subtract carry bits from decimal carry lookahead logic on MVL and MVR. All decimal adds and subtracts take 44ns as opposed to 22ns for binary operations. (dest MVEI)

2.2.3.8 DIAG ACT OUT (source For MVL=MVRL, Source For MVR=ESEJ) -

Serial data input for the MVL,MVR portions of the EBOX1 active scan path. (dest MVLF,MVRF)

2.2.3.9 DIAG CLK ACT (source For MVL=CLKB, Source For MVR=MVLM) -

The clock for the EBOX1 active scan path. This clock is used to parallel load or serial shift the 100141s on MVL and MVR. It is inclusively or'ed with system clocks which normally parallel load these registers. (dest MVLM, MVRM)

2.2.3.10 DIAG SH ACTIVE EBOX (source CLKG) -

When this signal is high, the scan path is in serial shift mode; when low, the scan path is in parallel load mode. (dest SCAH, CRAH, SHLM, SHRM, MVLF, MVRF, ESEI)

2.2.3.11 EXT 00,18 TO SELF, EXT 00,18 TO MVR (source MVLH) -

These bits are copies of SEL OP2 bits 00 and 18. Under R MUX control (EBOX microword bits 13:16), the value of SEL OP2 bit 00 or 18 may be gated into either the left or right half of the R MUX. See EBOX MICROCODE CONTROL SPECIFICATION under R/Y REG RD. (dest MVL6, MVR6)

2.2.3.12 FORCE ALU ERR (source CRAA) -

This signal is strobed on by diagnostic controls at CRA. MVL and MVR ALU err latches will be set by turning FORCE ALU ERR on and leaving it on while causing CLK 1 GATED A (MVLM, MVRM) to occur. (dest MVLL, MVRL)

2.2.3.13 FORCE LO L CLK (source IDRH) -

This signal is strobed by diagnostic controls at IDR. It allows results from the ALU or MOVE path to be gated through the L MUX while EBOX microcode does not advance. Thus a single micro-word causing a MOVE or ALU operation to occur may be issued, and results may be routed into and scanned at L REG. Normally, L MUX may not be clocked until CLK 1 of the machine cycle following the MOVE or ALU operation. (dest MVLM, MVRM)

2.2.3.14 GLOBAL (source CRAH) -

Under CARRY CONTROL, GLOBAL may be used to determine whether The main ALU will be in 36 or 18 bit mode. If GLOBAL is on, it may be selected to inhibit a carry into bit 17 of the ALU. (dest MVLI)

2.2.3.15 G Ø9-13, 14-17, 18-22, 23-26, 27-31, 32-35 (source MVEJ) -

These are the carry generates which are used by MVL and MVR to determine the carry-lookahead selection of the F MUX. (dest MVLI, MVRI)

2.2.3.16 L EARLY B (source CRA7) -

Gates the clocking of F BUS and RF BUS latches. In a 44ns microcycle, under L EARLY EVEN and L EARLY ODD control from EBOX microcode, F and RF may be clocked in the even cycle (first half), odd cycle (second half), or both even cycle and odd cycle.

2.2.3.17 LSW Ø:1 Uw Ø9:1Ø (source ESEB, ESEC) -

Control the selection of bits ØØ:17 and associated parity of the SW MUX (also called PASS DP). See EBOX MICROCODE CONTROL SPECIFICATION under SW/ X REG RD. (dest MVE1-5)

2.2.3.18 L UW 23 (source ESED) -

Controls the selection of RF BUS or F BUS through the L MUX. See EBOX MICROCODE CONTROL SPECIFICATION under L. (dest MVEL)

2.2.3.19 MAS AC MUX Ø:3 (source SHLF, SHRF) -

High-order 4 bit address to the AC BACKUP RAMs. The AC BACKUP is a copy of the 256 word master AC set and scratch pad at SHL and SHR. It is used by the console during hardware retry to recover data in the event of a MAC parity error. (dest MVEF)

2.2.3.2Ø OP1=1S (source CRAF) -

A signal resulting from an EBOX special function field micro-order, causes a set of SEL OP1 data bits and a reset of SEL OP1 parity bits. It allows SEL OP2 to pass through to the COMP BUS and-gates for compare to Ø, -1, or +1. See EBOX MICROCODE CONTROL SPECIFICATION

under SP FUNCTION. (dest MVE1)

2.2.3.21 OP1 UW Ø7 (source ESEB, ESEC) -

From EBOX microcode, controls selection into the SEL OP1 MUX of either AF MUX direct or AF MUX with halfwords swapped. See EBOX MICROCODE CONTROL SPECIFICATION under OP1/WR REG FI. (dest MVE1-4)

2.2.3.22 OP2=1S (source CRAF) -

A signal resulting from a EBOX special function field micro-order, causes a set of SEL OP2 data bits and a reset of SEL OP2 parity bits. It allows SEL OP1 to pass through to the COMP BUS and-gates for compare to Ø, -1, or +1. See EBOX MICROCODE CONTROL SPECIFICATION under SP FUNCTION (dest MVEH)

2.2.3.23 OP2 BUS (source IDL1-5, IDR1-5) -

Timing: lo-order bit must be valid T2+1.Ø to T2+12.7 /LU hi-order bit must be valid T2+2.Ø to T2+13.7 /LU 36 bit data path from IBOX to EBOX. It may be selected into the X1 MUX and thus into SEL OP2 via the EBOX microcode X1 field control bits. (dest MVE1-5)

2.2.3.24 P Ø9-13, 14-17, 18-22, 23-26, 27-31, 32-35 (source MVEJ) -

These are the carry propagates which are used by MVL and MVR to determine the carry-lookahead selection of the F MUX. (dest MVLI, MVRI)

2.2.3.25 PHASE Ø:3 TO MVL, MVR (source CLK7-A) -

Clocks to MVL and MVR from the CLK module. See CLK MODULE SPECIFICATION in chapter 1. (dest MVEM)

2.2.3.26 R Ø:3 UW 13:16 (source ESEB, ESEC) -

From EBOX microcode, controls the selection of data inputs to the R MUX. See EBOX MICROCODE CONTROL SPECIFICATION under R/ Y REG RD. (dest MVE6)

2.2.3.27 RF UW 1Ø4 (source CRAD) -

From EBOX microcode, controls selection of RF MUX to be R BUS if off, or A BUS, if on. See EBOX MICROCODE CONTROL SPECIFICATION under RF. (dest MVEL)

2.2.3.28 RSW Ø:1 UW 11:12 (source ESEB, ESEC) -

Control the selection of bits 18:35 and associated parity of the SW MUX (also called PASS OP). See EBOX MICROCODE CONTROL SPECIFICATION under SW/ X REG RD. (dest MVE1-5)

2.2.3.29 SYS RESET (source CLKB) -

Sets SEL OP1 and SEL OP2 to 1s with good parity, resets Carry Out, F BUS, L BUS, and RF BUS latches to Øs with good parity. (dest MVEM)

2.2.3.3Ø WR BK Ø:3 (source SCAH) -

Timing: must be valid TØ+3.8 to tØ+8.5 /LU Hi-order 4 bits of address to AC BACKUP RAMs. The AC BACKUP is a copy of the 256 word Master AC set and scratch pad at SHL and SHR. It is used by the console during hardware retry to recover data in the event of a MAC parity error. (dest MVEF)

2.2.3.31 WRITE BKUP (source ESEJ) -

Control signal caused by EBOX microcode STORE field will enable a write of L REG data into the MAC backup RAMs. (dest MVEF)

2.2.3.32 X BUS -2, -1, ØØ:35, 4P (source SHL7, SHR7) -

Data path from the SHIFT, AC, and REG FILE portion of the EBOX. X BUS may be gated into the X1, Y1 or AF MUXes under microcode or forwarding control. During the 1st cycle of an instruction it may only be selected into the X1 MUX if a match has occurred between the AC value of the OP2 address (if it represents an AC) of the current instruction and the AC address of a STORE AC operation in progress. X BUS -2 and -1 are used for EBOX multiply operations. (dest MVE1-5)

2.2.3.33 X1 EN (source IDLE) -

From ISET microcode in the IBOX, is used to force the output of the X1 MUX to zeros during the 1st cycle of selected instructions. See IBOX MICROCODE CONTROL SPECIFICATION, ISET, under X1 DISABLE. (dest MVE1)

2.2.3.34 X1 0:1 UW 37:38 (source ESEK) -

Control field from EBOX microcode which determines the selection of inputs to the X1 MUX. See EBOX MICROCODE CONTROL SPECIFICATION under X1. (dest MVE1-5)

2.2.3.35 Y BUS 00:35, 4P (source SHLA, SHRA) -

Data path from the SHIFT, AC, REG FILE portion of the EBOX. Y BUS may be gated into the Y1 and AF MUXes under microcode or forwarding control. (dest MVE1-5)

2.2.3.36 Y1 EN (source IDLE) -

From ISET microcode in the IBOX, is used to force the output of the Y1 MUX to zeros during the 1st cycle of selected instructions. See IBOX MICROCODE CONTROL SPECIFICATION, ISET, under Y1 DISABLE. (dest MVE1)

2.2.3.37 Y1 0:1 UW 39:40 (source ESEK) -

Control field from EBOX microcode which determines the selection of inputs to the Y1 and AF MUXes. See EBOX MICROCODE CONTROL SPECIFICATION under Y1. (dest MVE 1-5,L)

2.2.4 Interface: Outputs From MVL,MVR Modules

2.2.4.1 CARRY OUT (source MVL8) -

Carry out of bit 0 of the main ALU, is clocked by a delayed clock 2. Carry out goes to CRA as a Next Address dispatch input, and is then sent from CRAI to SCAE under the name of CRY OUT as an input to PC FLAGS CRY0 and CRY1. It is also latched by the following clock 1 at CRAI and sent to SHRJ under the name of CARRY OUT OF ALU as an L2MQ REG input. (dest CRAG, CRAI)

2.2.4.2 COMP ØØ, Ø9 (source MVE6) -

COMP ØØ is the "and" of SEL OP1 bit ØØ and SEL OP2 bit ØØ. COMP Ø9 is the "and" of SEL OP1 bit Ø9 and SEL OP2 bit Ø9. The "and" of the two bits is gated to IBOX jump determination logic if the EBOX microcode R/ Y REG RD field selects the COMPARE BUS into the R MUX. (dest IDLI)

2.2.4.3 COMP Ø1-Ø8, 1Ø-17, 18-26, 27-35 EQ (source MVEM) -

The equal result of a compare of the COMP BUS to Ø, -1, or +1. These signals are sent to IBOX jump determination logic. The value being compared against is determined by the EBOX microcode COMP field. (dest IDLI)

2.2.4.4 COMP Ø1-Ø8, 1Ø-17, 18-26, 27-35 G (source MVEM) -

These signals denote that a segment of the COMP BUS is greater than the value Ø, -1, or +1. They are sent to IBOX jump determination logic. The value being compared against is determined by the EBOX microcode COMP field. (dest IDLI) h13 COMP Ø1-Ø8, 1Ø-17, 18-26, 27-35 LS (source MVEM)

These signals denote that a segment of the COMP BUS is less than the value -1 or +1, as determined by the EBOX microcode COMP field. They are sent to IBOX jump determination logic. (dest IDLI)

2.2.4.5 CRY To 35 (source MVLI) -

From carry control logic on MVLI, is used for carry lookahead there and is the sent to MVRI for the same purpose. (dest MVRI)i

2.2.4.6 DCAR Ø8, 17, 35 (source MVEI) -

Decimal add/subtract carry bits from decimal carry lookahead logic on MVL and MVR. All decimal adds and subtracts take 44ns as opposed to 22ns for binary operations. (dest MVEI)

2.2.4.7 DIAG CLK ACT MVL, MVR (source MVEM) -

This clocked is issued under diagnostic control to shift or parallel load the EBOX1 active scan path. (from MVL, dest=MVRM, from MVR, dest=ESEI)

2.2.4.8 EXT ØØ,18 TO SELF, EXT ØØ,18 TO MVR (source MVLH) -

These bits are copies of SEL OP2 bits ØØ and 18. Under R MUX control (EBOX microword bits 13:16), the value of SEL OP2 bit ØØ or 18 may be gated into either the left or right half of the R MUX. See EBOX MICROCODE CONTROL SPECIFICATION under R/Y REG RD. (dest MVL6, MVR6)

2.2.4.9 F BUS (source MVE8, 9, A) -

F BUS is the 38 bit result of the main ALU. It is clocked at a delayed phase 2. Some of the F BUS high-order bits are sent to CRA as Next Address dispatch inputs. F BUS -Ø1 and F BUS -Ø2 are latched at CRAI with a Clock 2 and sent to SHLH D MUX inputs as F REG -Ø1 and F REG -Ø2. F BUS ØØ and F BUS -Ø1 are sent to SCAE for generation of the overflow indication. (dest CRAI, CRAG, SCAE)

2.2.4.10 G Ø9-13, 14-17, 18-22, 23-26, 27-31, 32-35 (source MVEJ) -

These are the carry generates which are used by MVL and MVR to determine the carry-lookahead selection of the F MUX. (dest MVLI, MVRI)

2.2.4.11 L BUS (source MVED, E, F) -

The L BUS is the 36 bit result data path of the EBOX. It is used to send data to memory or ACs to be stored, to send a memory address and qualifiers to the IBOX, and to re-circulate data back through the SHL, SHR data paths. It is selected from the F BUS, R BUS, or A BUS, under EBOX L and RF control bits. (hi dest=SHLJ, SHRJ, FPA, lo dest=IDLA, IDRG, MDPN)

2.2.4.12 MVL, MVR ALU ERR (source MVEL) -

Indicates that the propagates and generates of the duplicate ALUs were not equal. Once set, it will hold until CLEAR ERRS is issued under diagnostic control. It is also latched and held at the register inputs to the FRU logic at MID. Providing no other higher priority failure has occurred at the same time, ALU ERR will set an FRU code indicating an MVE failure. (dest MID1)

2.2.4.13 MVL, MVR BACKUP ERR (source MVEK) -

Indicates that bad parity was detected at the output of the MAC backup RAMs during a non-write cycle. (dest CFL)

2.2.4.14 MVL, MVR INPUT ERR (source MVEK) -

Indicates that bad parity was detected at SEL OP1 or SEL OP2. It will cause a freeze of the SEL OP1, SEL OP2 latches and ALU outputs. It is latched and held at the register inputs to the FRU logic at MID. If an X BUS, Y BUS or OP2 BUS, or other higher priority failure has not occurred at the same time, the FRU code will indicate an MVE failure. (dest MID1)

2.2.4.15 MVL, MVR OUTPUT ERR (source MVEK) -

Indicates that bad parity was detected on the L BUS. It will cause a freeze of the L BUS by blocking further clocks to the L MUX. It will also block writes to the MAC backup RAMs. It is sent to MID where it is latched and held at the register inputs to the FRU logic. Providing no other higher priority failure has occurred at the same time, the FRU code will indicate an MVE failure. (dest MID1)

2.2.4.16 OVERFLOW (source MVL8) -

Overflow from the main ALU to set OV and CRY1 flags under FLAG FIELD control. (dest SCAE)

2.2.4.17 P Ø9-13, 14-17, 18-22, 23-26, 27-31, 32-35 (source MVEJ) -

These are the carry propagates which are used by MVL and MVR to determine the carry-lookahead selection of the F MUX. (dest MVLI, MVRI)

2.2.4.18 ØØ-Ø8, Ø9-17, 18-26, 27-35 EQUAL (source MVEM) -

The compare result of F BUS to -1, +1, or Ø according to the setting of the EBOX microcode COMP field. These signals go to CRA as Next Address dispatch inputs. (dest CRAI)

2.3 SHL,SHR MODULES (CPU BACKPANEL SLOTS 3,4)

2.3.1 SHL,SHR Functionality

2.3.1.1 SHIFT/AC/REG Path - The SHIFT/AC/REG path is contained on two identical modules, SHL and SHR. It consists of a 44ns loop through a SHIFT MATRIX, AC sets and register files, and muxing necessary to route the outputs of all of these into either the SHIFT MATRIX or the MOVE/ALU path.

2.3.1.2 Shift Operation - The SHIFT MATRIX is a 72 bit input, 36 bit output left shifter with a maximum shift value of 36 (100100). It's hi-order input comes from the X REG, loaded from the X BUS at CLK 2 gated by not L EARLY A. It's lo-order input comes from the Y REG, loaded from the Y BUS at CLK 2 gated by not L EARLY A. The shift output is available at the Z MUX 30ns after the clocking of the X and Y REGs, at CLK 0 gated by L EARLY B.

2.3.1.3 REGISTER FILE OPERATION - Two 16 word register files are present on SHL and SHR (SHL= bits 0:17 and SHR= bits 18:35). These are used as temporary storage files which are identical in content. The write address is determined by control bits 33:36; write timing is determined by L EARLY. A write will occur in the second or subsequent cycles if bit 59 is on. X REG RD is controlled by bits 9:12 and Y REG RD is controlled by bits 13:16.

2.3.1.4 AC/MASTER AC OPERATION - Two copies of the current AC set are kept on SHL and SHR. The X AC read port is addressed by bits 32:35 of the EACALC in the last cycle of an instruction, so that the X MUX may select between the data of an AC store in progress <if conflict compare> and the output of the X AC for an operation in the first cycle of the next instruction. d The X1 MUX at the ALU will select between a memory operand <if EACALC is a memory location> or the X BUS <if EACALC is an AC>, or R or RF BUS if forwarding from a previous cycle's results. The Y AC read port is addressed by instruction bits 14:17 in the last cycle of an instruction, so that the Y MUX may select between the data of an AC store in progress <if conflict compare> and the output of the Y AC for an operation in the first cycle of the next instruction.

The X AC and Y AC addressing is determined in second and subsequent cycles by control bits 33:35.

The MASTER AC set is a 256 word array of 128 AC sets and 128 temporary registers. It may read or write every 22ns, as opposed to the current AC sets which may read and write every 22ns.

The addressing of the MASTER AC set is determined by control bits 33:36 Write enable timing is controlled by L EARLY and STORE CONTROL. The STEP CTRS microorder will cause the MASTER AC to be read followed by an increment of its 4 lo-order bits by 1. Two increments of one will occur during the 40ns period containing STEP CTRS. One microcycle later,under dly control, the copies of the selected AC set will be written with the data from the MASTER AC, followed by an update of the AC sets' address +1 <this occurs twice in the second microcycle>. The 16 words of each AC set may thus be loaded at a rate of 1 per 22ns, with the data travelling from the MASTER AC set through the Z MUX, X MUX, X1 MUX, R MUX, RF MUX, L MUX, L1 MUX, and the MQ MUX.

2.3.2 Error Control

Parity is generated at the output of the SHIFT MATRIX. Four parity bits are carried with all 36 bit wide data paths of SHL and SHR. Parity is modified at the D and X MUXes to compensate for the 1 and 2 bit shifts. Parity checking occurs at the output of the MAC REG, X REG, and Y REG.

A parity error at the MAC REG will set R MAC REG ERR (for SHR) and L MAC REG ERR (for SHL) at the CLK 2 following the CLK Ø to MAC REG. This error condition will a) block the next clock to the MAC REG, and b) signal to error control logic at INX to stop all system clocks starting with CLK Ø 33ns later.

A parity error at X REG or Y REG will set R REG OR AC ERR (for SHR) and L REG OR AC ERR (for SHL) at the CLK Ø following the CLK 2 to X AND Y REG. This error condition will a) block the next clock to the X AND Y REGs, and b) signal to error control logic at MID to stop all system clocks at CLK Ø 22ns later.

System reset will reset L1 REG, X REG FILE, Y REG FILE, and Z MUX to zero with good parity. Error reset will reset the error latches 'MAC OR Z ERR' and 'REG OR AC ERR'.

2.3.3 Interface: Inputs To SHL, SHR Modules

2.3.3.1 AC ADR FLD Ø9:12 (source INX1, IDLE) -

At the LAST CYCLE of an instruction, a new SELI is generated which will read the AC fields of the two possible successor instructions to that SELI. During the execution cycles of SELI (1st cy through last cy) the AC ADR FLD will contain the AC FIELD of the predicted successor instruction. At LAST CYCLE of SELI, AC ADR FLD Ø9:12 will be clocked into the Y AC RD ADR latches at SHL and SHR. (dest SHLF, SHRF)

2.3.3.2 CARRY OUT OF ALU (source CRAI) -

Latched at clock 1 on CRA, is the carry out of bit Ø of the previous cycle's main ALU operation at MVL/MVR. At SHRI it is used as an input to the L2MQ register, and will be shifted into L2MQ REG 35 at the following phase 3 if an L2MQ shift left of 1 is specified by the D FIELD of the EBOX microcode.

2.3.3.3 D Ø:2 UW 1ØØ:1Ø2 (source CRAC) -

Controls shifting/hold of L2MQ and D MUX input selection. See EBOX MICROCODE CONTROL SPECIFICATION under D FIELD. (dest SHLJ, SHRJ)

2.3.3.4 DIAG ACT OUT SHL (source SHL5) -

Serial scan data of the EBOX2 diagnostic shift path. (dest SHRC)

2.3.3.5 DIAG CLK ACT EB2 (source CLKB) -

The clock for the EBOX2 active scan path. This clock is used to parallel load or serial shift the 1ØØ141s on the SHR module. It is inclusively or'ed with system clocks which normally load these registers. (dest SHRL)

2.3.3.6 DIAG CLK ACT SHR (source SHRL) -

The clock for the EBOX2 active scan path. Starting at the CLK module as DIAG CLK ACT EB2, this clock passes from SHR to SHL to SCA to CRA. It is used by diagnostic control to parallel load or serial shift the scan path registers. It is inclusively or'ed with the system clocks which normally load these registers. (dest SHLL)

2.3.3.7 DIAG SH ACTIVE EBOX (source CLKG) -

When this signal is high, the scan path is in serial shift mode; when low, the scan path is in parallel load mode. (dest SCAH, CRAH, SHLM, SHRM, MVLF, MVRF, ESEI)

2.3.3.8 D MUX ØØ A (source SHL7) -

Sign bit copy from SHL to be loaded into X register file sign position on SHR.(dest SHR7)

2.3.3.9 EN AC WRITE A,B (source ESEJ) -

Write control for the current AC sets at SHL and SHR. When active, a write to the ACs will occur in the following cycle. (dest SHL8, SHR8)

2.3.3.10 EN MAS AC (source ESEJ) -

Write control for ths Master AC sets at SHL and SHR. When active, a write to the Master ACs will occur in the following cycle. (dest SHLD, SHRD)

2.3.3.11 ESTEP ACT OUT SCA (source SCAB) -

Diagnostic scan path serial output from SCA. (dest SHLC)

2.3.3.12 F REG -1,-2 (source CRAI) -

F BUS is the 38 bit result of the main ALU. It is clocked at a delayed phase 2. Some of the F BUS high-order bits are sent to CRA as Next Address dispatch inputs. F BUS -Ø1 and F BUS -Ø2 are latched at CRAI with a Clock 2 and sent to SHLH D MUX inputs as F REG -Ø1 and F REG -Ø2. F BUS ØØ and F BUS -Ø1 are sent to SCAE for generation of the overflow indication. (dest CRAI, CRAG, SCAE)

2.3.3.13 HOLDOFF TO SHL, SHR (source MIDA) -

With WR 3 UW 36 off, Holdoff to SHR will cause VMA 32:35 to be selected into the 4 hi-order bits of the Master AC address. Holdoff to SHL, SHR will occur due to holdoffs which are the effect of EBOX waiting for a) IBOX to respond to a SPFN ICMD, b) an MBOX access to complete, represented by MBOX RESPONSE or ABORT CYCLE. (dest SHLF, SHRF)

2.3.3.14 L BUS (source MVED, E, F) -

The L BUS is the 36 bit result data path of the EBOX. It is used to send data to memory or ACs to be stored, to send a memory address and qualifiers to the IBOX, and to re-circulate data back through the SHL, SHR data paths. It is selected from the F BUS, R BUS, or A BUS, under EBOX L and RF control bits. (hi dest=SHLJ, SHRJ, FPA, lo dest=IDLA, IDRG, MDPN)

2.3.3.15 L EARLY ON A,B (source CRA7) -

Controls the clocking of Z MUX, X REG, and Y REG on SHL and SHR. If the source for Z MUX is the shift matrix, L EARLY on will block the clocking of Z at clock 3 of the next machine cycle. If the source for Z MUX is the MAC set, L EARLY has no effect on the clocking of Z. Step AC Adr Dlyd 2, Last Cycle, or "not" L EARLY ON enables the clocking of X and Y REGS.(dest SHLH, SHRH)

2.3.3.16 LAST CYCLE T1 C,D (source CRAI) -

Last cycle control from microcode causes clocking of Y AC read address for the next instruction. (dest SHRF,SHLF)

2.3.3.17 L1 REG 16:18 (source SHRI, SHLI) -

L1 REG bits 16:17 go from SHL to DMUX 18:19 at SHR for a shift right of one or two positions. L1 REG bit 18 goes from SHR to DMUX 17 input at SHL for a shift left of one position. (dest SHRG, SHLG)

2.3.3.18 L2MQ REG 16:18 (source SHRI, SHLI) -

L2MQ REG bits 16:17 go from SHL to SHR L2MQ shift input 18 for a shift right of one or two positions. L2MQ REG bit 18 goes from SHR to L2MQ 17 shift input for a shift left of one position.(dest SHRI, SHLI)

2.3.3.19 MAC MUX 1:3 (source SCAH) -

AC block selection if MAS AC ADR Ø is off. MAC MUX 1:3 is selected from #FLD 11:13, current AC block, or previous AC block. If MAS AC ADR Ø Ø is on, 128 scratch pad locations are accessible by #FLD 1Ø:17. (dest SHRF,SHLF) SHLF)

2.3.3.20 MAC TO Z REG (source CRAF) -

EBOX microcode SPFN 3 causes selection of the shift matrix to the Z MUX via "not" MAC TO Z REG. At all other times MAC is selected to Z. (dest SHR3, SHL3)

2.3.3.21 MAS AC ADR Ø (source SCAH) -

If UW 36 "sel #FLD" is on, #FLD 1Ø is gated to MAS AC ADR Ø, the hi-order bit of the master AC address.(dest SHRF, SHLF)

2.3.3.22 MQ UW 1Ø3 (source CRAD) -

Controls the mode of operation of the L2MQ REG. If off, the L2MQ performs a parallel load with data from the L1 REG. If on, the L2MQ shifts per D MUX control (if the D MUX performs a non shift operation, the L2MQ holds it's value.(dest SHRJ, SHLJ)

2.3.3.23 OP2 AC ADR 32:35 (source CFL) -

The address of the X AC set supplied by the IBOX.(dest SHRE, SHLE)

2.3.3.24 PHASE Ø:3 TO SHL, SHR (source CLKC,D,E,F) -

Clocks to SHL and SHR from the CLK module. See CLK MODULE SPECIFICATION in chapter 1.(dest SHRL, SHLL)

2.3.3.25 PRESET AC LEFT, RIGHT (source CLKG) -

Console controlled clear of the register files which contain the current AC set. Used for EBOX initialization.(dest SHR8, SHL8)

2.3.3.26 REG WR UW Ø7 (source ESEB,C) -

Controls the write of the X and Y REG files in conjunction with L EARLY. See EBOX MICROCODE CONTROL SPECIFICATION under OP1/WR REG FILE. (dest SHR7,SHL7)

2.3.3.27 RESET SHL, SHR ERROR (source CLKG) -

Console controlled reset of the error latches on SHL and SHR. (dest SHRL, SHLL)

2.3.3.28 S BUS 0:5 (source SCA9) -

The value indicating the number of positions the shift matrix is to shift the X,Y REG doubleword to the left. (dest SHR3, SHL3)

2.3.3.29 SCAD 00:11, P (source SCA8) -

Path for merging hi-order bits into the data path via the D MUX, usually used to merge floating point exponents.(dest SHLH)

2.3.3.30 STEP AC ADR (source ESEJ) -

Caused by STORE CONTROL field decode 7, this signal is used to step AC addresses while a block of ACs is being loaded from the MAC set to the current AC set. (dest SHRJ, SHLJ)

2.3.3.31 SYSTEM RESET (source CLKG) -

Console controlled reset of the data paths on SHL and SHR. (dest SHR1, SHL1)

2.3.3.32 VMA 32:35 (source IDRC,D) -

Lo-order bits of the virtual memory address; is used to access the ACs in the event that a VMA access of the MBOX represents an AC address.(dest SHRE, SHLE)

2.3.3.33 WR 0:3 UW 33:36 (source ESED, CRAE) -

Write address for the X and Y REG files. See EBOX MICROCODE CONTROL SPECIFICATION UNDER "WR".(dest SHRF, SHLF)

2.3.3.34 X REG 18:35 (source SHR5) -

Input from SHR to that portion of the SHIFT MATRIX on SHL. (dest SHL4)

2.3.3.35 X Ø:2 UW 41:43 (source CRAC) -

EBOX microcode control of the X MUX selects. See EBOX MICROCODE CONTROL SPECIFICATION under "X". (dest SHR6,J, SHL6,J)

2.3.3.36 XRD Ø:3 UW Ø9:12 (source ESEB,ESEC) -

Read address of the X REG file. See EBOX MICROCODE CONTROL SPECIFICATION under "SW/XRD". (dest SHRA, SHLA)

2.3.3.37 Y REG ØØ:17 (source (SHL5) -

Input from SHL to that portion of the SHIFT MATRIX on SHR. (dest SHR4)

2.3.3.38 Y Ø:1 UW 44:45 (source CRAC) -

EBOX microcode control of the Y MUX selects. See EBOX MICROCODE CONTROL SPECIFICATION under "Y". (dest SHR9,SHL9)

2.3.3.39 YRD Ø:3 UW 13:16 (source ESEB,ESEC) -

Read address of the Y REG file. See EBOX MICROCODE CONTROL SPECIFICATION under "R/YRD" .(dest SHRA, SHLA)

2.3.3.40 # FLD ØØ:17, 2P (source ESEE,ESEF) -

EBOX microcode input to the data path via the X MUX. See EBOX MICROCODE CONTROL SPECIFICATION under "#". (dest SHR6, SHL6)

2.3.4 Interface: Outputs From SHL, SHR Modules

2.3.4.1 AC WR ADR MUX 0:3 (source SHRE, SHLE) -

(dest CFL, IDRG)

2.3.4.2 D MUX 00 A (source SHL7) -

Sign bit copy from SHL to be loaded into X register file sign position on SHR.(dest SHR7)

2.3.4.3 DIAG ACT OUT SHL (source SHL5) -

Serial scan data of the EBOX2 diagnostic shift path. (dest SHRC)

2.3.4.4 DIAG CLK ACT SHL (source SHLM) -

The clock for the EBOX2 active scan path. This clock is used to parallel load or serial shift the 100141 shift regs on SCA. It is 'or-ed' with system clocks which normally parallel load these registers. (dest SCAH)

2.3.4.5 L1 REG 16:18 (source SHRI, SHLI) -

L1 REG bits 16:17 go from SHL to DMUX 18:19 at SHR for a shift right of one or two positions. L1 REG bit 18 goes from SHR to DMUX 17 input at SHL for a shift left of one position. (dest SHRG, SHLG)

2.3.4.6 L2MQ REG 16:18 (source SHRI, SHLI) -

L2MQ REG bits 16:17 go from SHL to SHR L2MQ shift input 18 for a shift right of one or two positions. L2MQ REG bit 18 goes from SHR to L2MQ 17 shift input for a shift left of one position.(dest SHRI, SHLI)

2.3.4.7 MAC ERROR (source SHRL,SHLL) -

(dest MID2)

2.3.4.8 MAS AC MUX Ø:3 (source SHLF, SHRF) -

High-order 4 bit address to the AC BACKUP RAMs. The AC BACKUP is a copy of the 256 word Master AC set and scratch pad at SHL and SHR. It is used by the console during hardware retry to recover data in the event of a MAC parity error. (dest MVEF)

2.3.4.9 VMA 32:35 A,B -

2.3.4.1Ø X BUS -2, -1, ØØ:35, 4P (source SHL7, SHR7) -

Data path from the SHIFT, AC, and REG FILE portion of the EBOX. X BUS may be gated into the X1, Y1 or AF MUXes under microcode or forwarding control. During the 1st cycle of an instruction it may only be selected into the X1 MUX if a match has occurred between the AC value of the OP2 address (if it represents an AC) of the current instruction and the AC address of a STORE AC operation in progress. X BUS -2 and -1 are used for EBOX multiply operations. (dest MVE1-5)

2.3.4.11 X REG 18:35 (source SHR5) -

Input from SHR to that portion of the SHIFT MATRIX on SHL. (dest SHL4)

2.3.4.12 X REG FILE 18:19 -

2.3.4.13 X,Y ERROR (source SHRL, SHLL) -

(dest MID2)

2.3.4.14 Y BUS ØØ:35, 4P (source SHLA, SHRA) -

Data path from the SHIFT, AC, REG FILE portion of the EBOX. Y BUS may be gated into the Y1 and AF MUXes under microcode or forwarding control. (dest MVE1-5)

2.3.4.15 Y REG 00:17 (source (SHL5) -

Input from SHL to that portion of the SHIFT MATRIX on SHR. (dest SHR4)

2.3.4.16 Z REG P00-08 -

2.4 SCA MODULE (CPU BACKPANEL SLOT 1)

2.4.1 SCA Functionality

2.4.1.1 Shift Control/Meters/Flags - The SCA module contains a 13 bit ALU path used for controlling the SHIFT MATRIX shift value and floating point exponents. Also on this module are the accounting meter, interval timer, PI and APR flags, PC flags, time base, and the 4 bit State reg.

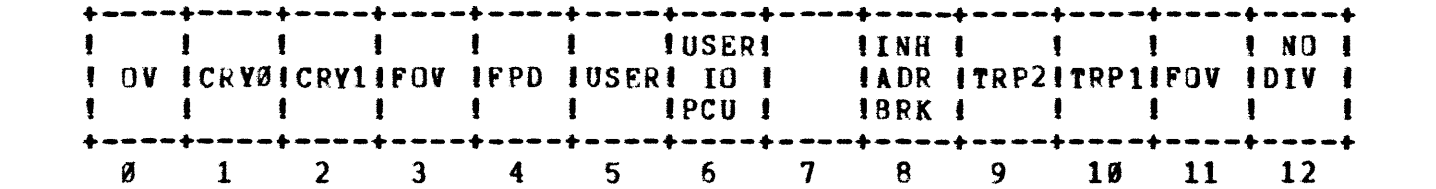
2.4.1.2 SCAD - The shift control ALU is a 12 bit binary add or subtract function receiving inputs from J and K MUXes. The output of the ALU is latched into the SCAD REG which provides the floating point exponent to the D MUX on the SHL module, and dispatches for branching under control of the next address switch. The output of the ALU is also latched into the SC REG, which may be held more than one cycle, and is used for branching and shift control purposes. The FE MUX receives inputs from either the ALU or the K MUX, and loads the FE REG at CLK2 gated by L EARLY A and FE UW bit 68 off. Both the SCAD REG and the SC REG are also clocked at CLK2 gated by L EARLY A.

2.4.1.3 1US Time Base - The TIME BASE consists of a 16 bit counter which updates at 1us intervals and is simultaneously readable and clearable by microcode. The INIT TIM micro-order with X REG 26 on presets the time base to zero while at the same time gating the last value of the counter (prior to preset) to the next microcycle's # field 2:17. INIT TIM with X REG 25 on enables the time base to update +1 from a synchronized 1us pulse. Conversely INIT TIM with X REG 25 off will block the update.

2.4.1.4 Interval Timer - The INTERVAL TIMER is a 12 bit count-up counter. The LD TIM micro-order sets a 12 bit interval into the INTERVAL register from X REG 24:35. LD TIM sets a flip latch to the value of X REG 21 to allow the timer to step up once every 10us (X REG 21 on and LD TIM turn the timer on). The interval is continuously compared to the count in the timer. A match will set the TIMER DONE flag. A carry out of counter will set the TIMER OVERFLOW flag. The INIT TIM micro-order sets the interval timer PIA from X REG 33:35. If the TIMER DONE flag is set, it will be steered into one of seven timer interrupt levels via the timer PIA decode. The timer interrupt levels, one of which may carry the timer interrupt, are sent to the interrupt system.

2.4.1.5 Flags - The eleven PC flags are located on the SCA module, and may be loaded from bits Ø:12 of the Y REG. They are read into the next # FIELD Ø:12 when PC FLAGS micro-order is issued.

PC FORMAT



The STATE REG is an 4 bit register of independently settable and resettable flags. STATE REG bits Ø:3 are set by # FIELD Ø:3, respectively, and reset by # FIELD 4:7, respectively, when the LD STATE micro-order is issued. The STATE REG bits are used to remember the status of microcode subroutines as portions of an instruction are completed. They may be read into # FIELD Ø:3 if the RD FLAGS micro-order is isued. STATE REG bits are primarily used as dispatches into the NA BUS under control of the NA SWITCH. The APR flags are a set of 4 flip-flops independently settable by asynchronous conditions external to EBOX operations. They may also be independently set or reset by the LD APR micro-order. The flags are labelled as follows:

flag	function
28	protocol done
29	console attention
3Ø	power failure
31	non-existent memory

The EN APR micro-order loads a register from X REG 24:31 which gates the respective APR flags into an "or" funnel. If any APR flag is on and enabled, an APR interrupt signal will be generated. The EN APR micro-order also loads the APR PIA from X REG 33:35. The APR PIA register selects the APR interrupt into the appropriate level (one through 7), which is then presented to the interrupt mechanism.

2.4.1.6 Interrupt Mechanism - The LD PI micro-order arms the JUPITER interrupt mechanism in the following manner: LD PI sets X REG BIT 28 into a latch which represents "interrupt system on" and allows an interrupt trap to occur at any Last Cycle. The microcode may also do a skip using interrupt request as an input to next address logic. This allows the interrupt to be taken in the middle of long instructions. Interrupt levels one through seven may be raised by IO, APR, or TIMER service requests. these levels are gated into a priority encoder by the "levels on" hardware register, a seven bit register loaded from X REG 29:35 by the LD PI micro-order. The output of the request priority encoder is compared with a latched encode of the current level in progress, which is loaded from X REG 21:27 by the LD PI micro-order. A compare is made between the current level and requested level. If the requested level is higher in priority (lower in number) than the current level in progress, an interrupt request line is generated. Interrupt request, Last Cycle, and Interrupt System On will generate an interrupt trap, forcing slow microcode to go to address 4Ø11 (octal), if no console trap is pending (console

trap is the only higher priority trap).

2.4.2 Interface: Inputs To SCA Module

2.4.2.1 ALU ØØ-17 = Ø (source CRAI) -

Signal which is the "and" at CRA of F BUS ØØ:Ø8 = zero and F BUS Ø9:17 = zero; causes the set of TRAP 2 flag if FLAG field decode 14. (dest SCAG)

2.4.2.2 CPU ID ØØ:13 (source CPU Backpanel) -

Individualized backpanel wiring of ECL high and low inputs which provide an identification number for the CPU. (dest SCAF)

2.4.2.3 CSL ATTN (source IOP) -

Interrupt from the console which enters the CPU as APR FLAG 2. (dest SCAI)

2.4.2.4 DIAG ACT OUT CRA (source CRA7) -

Serial output of EBOX2 diagnostic scan path from CRA. (dest SCA9)

2.4.2.5 DIAG CLK ACT SHL (source SHLM) -

The clock for the EBOX2 active scan path. This clock is used to parallel load or serial shift the 100141 shift regs on SCA. It is "or-ed" with system clocks which normally parallel load these registers. (dest SCAH)

2.4.2.6 DIAG SH ACTIVE EBOX (source CLKG) -

When this signal is high, the scan path is in serial shift mode; when low, the scan path is in parallel load mode. (dest SCAH, CRAH, SHLM, SHRM, MVLF, MVRF, ESEI)

2.4.2.7 EBOX AC Ø9:12 (source CFL) -

The AC field of the instruction currently in execution by the EBOX. (dest SCAH, MIDC, CRAG)

2.4.2.8 EBOX STEP B (source CFL Via CRAI) -

EBOX STEP from the IBOX indicates to the EBOX that the 1st cycle in progress is a valid operation, that all information needed to begin execution of the instruction was prefetched and presented properly to the EBOX by the IBOX. At CRAH it is used to gate the clock of the GLOBAL and PC SECT = Ø indicators. At SCAJ it is used to gate the update of the FLAG HISTORY BUFF address. (dest SCAJ)

2.4.2.9 EBOX XR 14:17 (source CFL) -

The index register field of the instruction currently in execution. It is valid during an EBOX routine at the IBOX. (dest SCA3)

2.4.2.10 FE CTL (source ESED) -

Controls clocking of the FE MUX at SCA6; from EBOX ucode bit 17. FE will clock in EVEN CYCLE if bit 17 on, in ODD CYCLE if bit 17 off. See EBOX MICROCODE CONTROL SPECIFICATION under "COMP". (dest SCA7)

2.4.2.11 FE Ø:1 UW 68:69 (source CRA7) -

Control clocking of the FE MUX. See EBOX MICROCODE CONTROL SPECIFICATION under FE FIELD. (dest SCA7)

2.4.2.12 FLAG Ø:3 UW 29:32 (source ESED, CRAD) -

The FLAG field determines the update of flags and counters resident on SCA. see EBOX MICROCODE CONTROL SPECIFICATION under "FLAG CONTROL". (dest SCAE)

2.4.2.13 FORCE 1ST TO SCA (source CFL Via CRAH) -

Force 1st will prevent the update of flags on SCA, and control the the setting of ODD and EVEN CYCLE latches at ESE and CRA. Force 1st indicates that the instruction is not valid, that the IBOX not yet completed the prefetch of information necessary for the EBOX to execute the current instruction. (dest SCA7)

2.4.2.14 IL 1:5 TO CPU (source IOP) -

Interrupt levels 1 through 5 from the IO PORTS. (dest SCAA)

2.4.2.15 INCREM TBC (source MIDB) -

1 micro-second interval signal which is synchronized to the CPU clocks by logic on MID. This signal causes the Time Base Counter update by +1. (dest SCAB)

2.4.2.16 INT TIM +1 (source MIDB) -

10 micro-second interval signal which is synchronized to the CPU clocks by logic on MID. This signal causes the update of the Interval Timer. (dest SCA8)

2.4.2.17 J 0:2 UW 71:73 (source CRAC) -

EBOX microcode control of the J MUX selects. See EBOX MICROCODE CONTROL SPECIFICATION under "J FIELD". (dest SCA3)

2.4.2.18 K 0:2 UW 74:76 (source CRAC) -

EBOX microcode control of the K MUX selects. See EBOX MICROCODE CONTROL SPECIFICATION under "K FIELD". (dest SCA4)

2.4.2.19 L EARLY ON (source CRA7) -

Gates the clocking of the FLAGS under the control of the FLAG FIELD. L EARLY ON will enable flags at SCAE and SCAG to set at clock 2 of the next machine cycle. (dest SCAB)

2.4.2.20 LAST CYCLE A (source ESEK) -

From EBOX microcode LAST CYCLE control, causes ODD CYCLE at SCA to synchronize with ODD CYCLE at ESE and CRA. (dest SCA7)

2.4.2.21 LD SPFN REG (source CRAF) -

Control to load the SPFN REG from the # FIELD. The SPFN REG is cleared at CLK 3 of Last Cycle. LD SPFN REG also gates # FIELD t-o enable other special functions not contained in the SPFN REG, such as FORCE PHYSICAL. See EBOX MICROCODE CONTROL SPECIFICATION under "SPFN". (dest SCAH)

2.4.2.22 OVERFLOW (source MVL8) -

Overflow from the main ALU to set OV and CRY1 flags under FLAG FIELD control. (dest SCAE)

2.4.2.23 PHASE Ø:3 TO SCA (source CLKC,D,E,F) -

Clocks to SCA from the CLK module. See CLK Module specification in chapter 1. (dest SCAK)

2.4.2.24 PREV EN A (source ESEJ) -

Control to select the previous AC block number to address the MAC and AMC backup. (dest SCAH)(dest SCAH)

2.4.2.25 PWR FAIL (source IOP) -

Indication of an AC power failure presented as an interrupt via APR FLAG 3. (dest SCAI)

2.4.2.26 RESET CRA ERROR (source CLKG) -

Console controlled clear of error latches on CRA and SCA modules. (dest SCA2, CRA7)

2.4.2.27 RESET SCA (source CLKG) -

Console controlled clear of flags, latches, and data paths on SCA. (dest SCA2)

2.4.2.28 SC CTL (source ESED) -

From COMP field bit 17. If UW 17 is on, SC REG will clock in EVEN CYCLE; if UW 17 is off SC REG will clock in ODD CYCLE. See EBOX MICROCODE CONTROL SPECIFICATION under 'COMP'. (dest SCA7)

2.4.2.29 SC Ø:2 UW 65:67 (source CRAC) -

EBOX microcode control of the SC MUX selection and the clocking of the SC REG. See EBOX MICROCODE CONTROL SPECIFICATION under 'SC FIELD'. (dest SCA7)

2.4.2.30 SCU ALU UW 77 (source CRAB) -

EBOX microcode control which determines the mode of operation of the SC ALU. If UW 77 is on SC ALU is in subtract mode; if UW 77 is off SC ALU is in add mode. (dest SCA7)

2.4.2.31 SEL #FLD (i (source CRAE) -

From UW bit 36. If on, cause selection of # FIELD 1Ø:13 to the MAC and MAC BACKUP hi-order 4 address bits from SCA. UW 36 on at SHR and SHL will cause selection of # FIELD 14:17 to MAC and MAC BACKUP lo-order 4 address bits. (dest SCAH)

2.4.2.32 SPFN LD BLK (source CRAF) -

EBOX microcode SPFN decode 2 to load the current and previous AC blocks. Current from Y REG Ø:2; Previous from Y REG 3:5. (dest SCAH)

2.4.2.33 SPFN LD IO REG (source CRAF) -

EBOX microcode SPFN decode 6 to load IO REG with control information for the IO PORTS. A strobe from CRA signals to the IOP that the IO REGISTER signals are valid. See EBOX MICROCODE CONTROL SPECIFICATION under 'SPFN'. (dest IOP)

2.4.2.34 SPFN LD PI (source CRAF) -

EBOX microcode SPFN decode 1 to load the PI system with control infromation. See EBOX MICROCODE CONTROL SPECIFICATION under 'SPFN'. (dest SCAA)

2.4.2.35 X REG 18, 28:35 (source SHR6) -

X Register bits from SHR. Bit 18 is used as a dispatch to CRA next address, as an input to the J MUX at SCA3, and as an input to the Shift Matrix at SHL. Bits 28:35 are used as inputs to the J MUX at SCA3, and the Shift Matrix at SHL. (dest CRAG, SCA1, SHL4)

2.4.2.36 Y REG ØØ:17 (source SHL6) -

Y Register bits from SHL. Bits Ø:3,13 are used as dispatches to the CRA next address; bits ØØ:17 are used as inputs to the Shift Matrix at SHR, and as inputs to K MUX and flags at SCA. (dest CRAG, CRAH, SHR4, SCA1)

2.4.2.37 # FLD ØØ:17 UW 8Ø:97 (source ESEE) -

Input to J MUX, K MUX, flags on SCA. # FLD bits ØØ:13 go to MID, latched by SPFN ICMD, as qualifiers for memory operations. # FLD bits 14:17 go to ISQ as the 4 lo-order bits of the IPUT microcode address of the 1st IPUT microword of an EBOX routine. See EBOX MICROCODE CONTROL SPECIFICATION under '# FLD' and 'SPFN ICMD'. (dest SCA2, MID8, ISQH)

2.4.3 Interface: Outputs From SCA Module

2.4.3.1 ALLOW RETRY (source SCAH) -

SPFN REG bit which, if off, indicates to the console, following a hardware error, that the instruction in execution had entered a region of microcode which precluded instruction retry. Allow retry is set on at Clock 3 of Last Cycle. (dest CRAG, INXC)

2.4.3.2 BLOCK STEP (source SCAH) -

Signal to the IBOX which will prevent the loading of a new SELI upon the occurance of the next Last Cycle, causing EBOX to remain in the current SELI. This allows instruction execution microcode at EBOX to be entered at the start of an instruction following the handling of setup due to page fail traps, the 'execute' instruction, etc. (dest

CRAF, CFL)

2.4.3.3 CLR JREAL (source SCAH) -

Causes the clear of the Jump Real latch at IDLI. For certain instructions, ISET (IDL) in 1st cycle is not capable of determining the true status of the Jump Real latch. In these cases, ISET will set JUMP REAL on unconditionally. Later, when the EBOX has determined the true state of the jump, CLR JREAL is used to clear the Jump Real latch if needed. (dest CFL)

2.4.3.4 DIAG CLK ACT SCA (SCAH) -

Diagnostic clock to the EBOX2 scan path. (dest CRAI)

2.4.3.5 DIS PAGE FAIL (source SCAH) -

SPFN reg bit gating of page fail interrupt during EBOX memory references. (dest CRAF)

2.4.3.6 DISP 09, NASW=27 (i (SCAK) -

EBOX microcode dispatch to next address. Function of Y REG 00, 06:17 < or = to zero. (dest CRAG)

2.4.3.7 EBOX NO JUMP (source SCAB) -

For FLAG FLD decode 10, is an indication that none of the flags CRY0, OV, CRY1, or FOV were cleared. It goes via CFL to IDLI where the JUMP REAL latch is reset. (dest CFL)

2.4.3.8 ECL BROADCAST (source SCAH) -

IO Register bit 31 loaded by SPFN LDIO, to IO PORTS. See EBOX MICROCODE CONTROL SPECIFICATION under 'SP FUNCTION, LD IO' (dest IOP)

2.4.3.9 ECL CTRL Ø:1 ENC (source SCAH) -

IO Register bits 28:29 loaded by SPFN LDIO, to IO PORTS. See EBOX MICROCODE CONTROL SPECIFICATION under "SP FUNCTION, LD IO". (dest IOP)

2.4.3.10 ECL S SEL Ø:3 (source SCAH) -

IO Register bits 32:35 loaded by SPFN LDIO, to IO PORTS. See EBOX MICROCODE CONTROL SPECIFICATION under "SP FUNCTION, LD IO". (dest IOP)

2.4.3.11 EN ADR BREAK (source SCAH) -

Signal from EBOX to MBOX to enable the address break feature during memory references. (dest MBOX)

2.4.3.12 ESTEP ACT OUT SCA (source SCAB) -

Diagnostic scan path serial output from SCA. (dest SHLC)

2.4.3.13 EXTEND INSN (source SCAH) -

SPFN indication to microcode next address bit 2, causing entry into microcode address space reserved for the extended instruction set. (dest ESE5)

2.4.3.14 FLAGS TO # FLD (source SCAE) -

Decode of flags field returned to ESE from SCA to cause the SCA flags MUX to be selected into the next microcycle's number field. See EBOX MICROCODE CONTROL SPECIFICATION under "FLAG CONTROL". (dest ESEI)

2.4.3.15 FLAGS ØØ:17, 2P (source SCAF) -

Output of FLAGS MUX from SCA to the next microcycle's number field. See EBOX MICROCODE CONTROL SPECIFICATION under "FLAG CONTROL". (dest ESEE)

2.4.3.16 FORCE PHY (source SCAH) -

Signal indicating that concurrent EBOX references to memory are represented by a physical address on the VMA. (dest CAMI)

2.4.3.17 INT TRAP ON (source SCAA) -

When turned off, prevents interrupts from occurring to the EBOX microcode. (dest CRAF)

2.4.3.18 INTERNAL SCA ERR (source SCA8) -

Indicates that an SC ALU or data path error has occurred at SCA; will cause an FRU code to be stored at MID, and a holdoff to EBOX and IBOX clocks. (dest MID2)

2.4.3.19 INTERRUPT REQ (source SCAA) -

Interrupt to the EBOX microcode from Software PI, IO PORTS, the APR system, or Timers. (dest CRAF)

2.4.3.20 INTR DIS Ø:2 (source SCAA) -

The level of the current INTERRUPT REQ from Software PI, IO PORTS, the APR system, or Timers. (dest CRAG)

2.4.3.21 J MUX > 44 (source SCA3) -

Branch bit (dispatch) to EBOX next address indicating that J MUX contains a value greater that the shift capability of the Shift Matrix at SHL, SHR. (dest CRAG)

2.4.3.22 MAC MUX 1:3 (source SCAH) -

AC block selection if MAS AC ADR Ø is off. MAC MUX 1:3 is selected from #FLD 11:13, current AC block, or previous AC block. If MAS AC ADR Ø Ø is on, 128 scratch pad locations are accessible by #FLD 1Ø:17. (dest SHRF,SHLF) SHLF)

2.4.3.23 MAGIC NUM 14:17 (source SCA2) -

FIELD bits used to increment the OP2 AC Address by required offset. See EBOX MICROCODE CONTROL SPECIFICATION under "WR CTRL". (dest CFL)

2.4.3.24 MAS AC ADR Ø (source SCAH) -

If UW 36 "sel #FLD" is on, #FLD 1Ø is gated to MAS AC ADR Ø, the hi-order bit of the master AC address.(dest SHRF, SHLF)

2.4.3.25 MASK TO 44 (source SCA9) -

Branch bit (dispatch) to EBOX microcode next address indicating that the current S BUS value was forced to 44 (shift of 36 positions) due to its selected input, SC REG or SC ALU, having a value greater that 44. (dest CRAG)

2.4.3.26 PCU/USER IO FLAG (source SCAG) -

PCU/USER IO FLAG goes to CRA for next address branching: to MID for control of previous context memory access and determination of VMA Ø5. (dest CRAI, MIDC)

2.4.3.27 RESET FROM CPU (source SCAH) -

IO REG bit 3Ø loaded by SPFN LDPI to IO PORTS. See EBOX MICROCODE CONTROL SPECIFICATION under "SP FUNCTION, LDPI". (dest IOP)

2.4.3.28 S BUS Ø:5 (source SCA9) -

The value indicating the number of positions the shift matrix is to shift the X,Y REG doubleword to the left. (dest SHR3, SHL3)

2.4.3.29 SC REG OVFL (source SCA5) -

Input to EBOX microcode next address branch (dispatch) indicating carry out of bit ØØ of the SC ALU, latched into the SC REG. (dest CRAG)

2.4.3.30 SC REG ØØ,Ø2,Ø3 (source SCA9) -

SC REG bits to EBOX microcode next address branch dispatch. (dest CRAG)

2.4.3.31 SC REG = Ø (source SCA9) -

Contents of SC REG = Ø, to EBOX next address dispatch. (dest CRAG)

2.4.3.32 SCAD ALU = Ø (source SCA8) -

Contents of SC ALU = Ø, to EBOX next address dispatch. (dest CRAG)

2.4.3.33 SCAD ØØ:11, P (source SCA8) -

Path for merging hi-order bits into the data path via the D MUX, usually used to merge floating point exponents.(dest SHLH)

2.4.3.34 TRAP 1/2 REQ (source SCAG) -

TRAP1 FLAG or TRAP2 FLAG was set during the previous instruction; an interrupt request to EBOX microcode at 1st cycle. (dest CRAF)

2.4.3.35 USER FLAG (source SCAG) -

USER FLAG to EBOX next address dispatch at CRA, and to previous context memory access selection and VMA Ø5 at MID. (dest CRAI, MIDC)

2.4.3.36 WR BK Ø:3 (source SCAH) -

Timing: must be valid TØ+3.8 to tØ+8.5 /LU Hi-order 4 bits of address to AC BACKUP RAMs. The AC BACKUP is a copy of the 256 word Master AC set and scratch pad at SHL and SHR. It is used by the console during hardware retry to recover data in the event of a MAC parity error. (dest MVEF)

2.4.3.37 Y REG Ø2:Ø5=Ø, 14:17=Ø (source SCAK) -

EBOX microcode next address branch inputs indicating the status of Y REG bits. (dest CRAG)

2.4.3.38 1MSEC REQ (source SCAB) -

1 milli-second interval interrupt to EBOX microcode requesting that the Time Base Counter be read and then cleared. (dest CRAF)

2.5 ESE,CRA MODULES (CPU BACKPANEL SLOTS 8,2)

2.5.1 ESE,CRA Functionality

EBOX control storage is located on the ESE and CRA modules. It contains 30 bits of fast and 105 bits of slow storage, plus the necessary output decodes and branching controls to operate the EBOX data paths and initiate operations at IBOX, MBOX, and FPA.

2.5.1.1 Ram Arrays - Two RAM arrays are present in EBOX control storage: a slow 4K deep array capable of cycling in 44ns, and a fast 512 word array which develops a short microword for control of the first cycle of each instruction. Each array is divided into 3 sections, early, intermediate, and late. Early control bits come from a section in the array that is directly powered by either the instruction code <fast RAMs in first cycle> or by a mux which selects instruction code or NA BUS <slow RAMs for second and subsequent cycles, respectively>. All early control bits are clocked at CLK 0. Intermediate control bits come from a section of the array that is skewed, at the address inputs and at the latched data outputs, 5 to 10 ns later than the early section. All intermediate control bits are clocked at either CLK 1 or CLK 2, depending upon the timing of the logic to be controlled. Late control bits come from a section which is further skewed by 5 to 10ns. All late control bits are clocked at either CLK 3 or CLK 2, depending the timing of the logic to be controlled.

The early, intermediate, and late control bits are deskewed into the CTL latches, which serve the dual purpose of holding that module's portion of the microword for parity checking, and providing a scan path input to the arrays for initial control storage load.

All 105 bits are addressable by instruction code for second cycle operation and next address path for third and subsequent cycle operation. Only 30 bits of fast RAM are used for first cycle operation, primarily to control the MOVE/ALU path and the store and flag setting controls for one cycle instructions. The control bits which do not have fast RAMs allocated default to zero in the first cycle. Control bits usable in the first cycle are:

0:3	ALU
4:6	CARRY CONTROL
7	OP1
8	LAST CYCLE
9:12	SW
13:16	R
17:18	COMP
19:21	STORE CONTROL
22	P ESE
23	L
24	MARK
25:26	SP FUNCTION
29:30	FLAG
33	WR

2.5.1.2 Next Address Generation - The NA BUS bits 0:11 are generated at the CRA module. The value of this bus is determined by the control bits of the Next Address Field and the NA SWITCH. The NA BUS addresses the slow array for control word readouts starting with the third cycle of an instruction.

2.5.1.3 Traps - The TRAP mechanism causes the microcode to land in one of eight addresses 401X if any of the following conditions (listed by priority) occurs: priority condition trap address 0(highest) hdwr error 4010 1 console.LC 4011 2 PI interrupt 4012 3 page fail.LC 4013 4 1MS trap.LC 4014 5 unused 4015 6 trap1 or trap2.last cycle 4016 7 IBOX trap.last cycle 4017

2.5.2 Error Control

Correct parity is stored into the ESE and CRA RAM arrays upon control storage loading from the console. As the microwords are read out, the portion of the word contained on each module is checked for odd parity (the early and intermediate CTL bits are relatched into registers clocked at the same time as the late CTL bits, so that the entire portion may be checked). If a parity check occurs, the CTL bits of that module are held, along with the NA DLY latch, which represents the address of the failing word (if the failure occured in third or subsequent cycle). If a parity error occurs in the first or second cycle, the console may read the instruction code of the failing instruction by stepping back the SEL I in the IBOX. The 34 fast RAMs are parity checked along with zeros (for unused positions) in the first cycle. A parity error detect will set ESE MICRO- CODE ERR or CRA MICROCODE ERR latches at CLK 2 following the late CLK 0 readout of a microcycle. These error latches will signal the error control logic at MID to stop IBOX, EBOX, and FPA system clocks starting with CLK 0, 30ns later.

System reset will reset all PC flags and STATE REG bits to Ø. Error reset will reset ESE and CRA MICROCODE ERR latches.

2.5.3 Interface: Inputs To ESE, CRA Modules

2.5.3.1 AC FLD N=N+1 (source INX1) -

An equal compare of bits Ø9:12 (AC field) of the next instruction to begin execution at the EBOX and the AC write address of the current instruction. According to L MUX control from EBOX microcode, AC FLD N=N+1 if active will allow selection of either F or RF MUXes into the Y1 MUX for 1st cycle operation. If inactive, Y MUX is selected into the Y1 MUX during 1st cycle. (dest ESEK)

2.5.3.2 AC REF TO ESE (source CFL) -

Indicates that the address on the VMA is an AC address as opposed to a memory address. True if an instruction address and VMA 18:31 = Ø, if an IBOX local fetch and VMA 18:31 = Ø, or if VMA Ø6:16 and 18:31 = Ø. At ESE gates the write to AC sets for a write memory (which is AC REF), and enables N+1 forwarding compare at CFL. (dest ESEJ)

2.5.3.3 AC=Ø (source CRAI) -

AC field of the EBOX instruction is zero. Causes block of write to AC if STORE FIELD 3. Also will block N+1 forwarding compare at CFL for STORE FIELD 3. See EBOX MICROCODE CONTROL SPECIFICATION under "STORE CONTROL". (dest ESEJ)

2.5.3.4 ALLOW RETRY (source SCAH) -

SPFN REG bit which, if off, indicates to the console, following a hardware error, that the instruction in execution had entered a region of microcode which precluded instruction retry. Allow retry is set on at Clock 3 of Last Cycle. (dest CRAG, INXC)

2.5.3.5 BLOCK STEP (source SCAH) -

Signal to the IBOX which will prevent the loading of a new SELI upon the occurance of the next Last Cycle, causing EBOX to remain in the current SELI. This allows instruction execution microcode at EBOX to be entered at the start of an instruction following the handling of setup due to page fail traps, the "execute" instruction, etc. (dest CRAF, CFL)

2.5.3.6 CARRY OUT (source MVL8) -

Carry out of bit Ø of the main ALU, is clocked by a delayed clock 2. Carry out goes to CRA as a Next Address dispatch input, and is then sent from CRAI to SCAE under the name of CRY OUT as an input to PC FLAGS CRYØ and CRY1. It is also latched by the following clock 1 at CRAI and sent to SHRJ under the name of CARRY OUT OF ALU as an L2MQ REG input. (dest CRAG, CRAI)

2.5.3.7 DIAG CLK ACT MVR (source MVRM) -

This clocked is issued under diagnostic control to shift or parallel load the EBOX1 active scan path. (from MVL, dest=MVRM, from MVR, dest=ESEI)

2.5.3.8 DIAG CLK ACT SCA (SCAH) -

Diagnostic clock to the EBOX2 active scan path. (dest CRAI)

2.5.3.9 DIAG CLK PAS 12 (source IDRM) -

Diagnostic clock for the CPU passive scan path.(dest CRAA)

2.5.3.1Ø DIAG CLK UW ADR (source CRAA) -

Console strobed clock to propagate the RETURN ADDRESS on CRA to the next address paths of CRA and ESE. (dest ESEK)

2.5.3.11 DIAG CMD ENABLE (source CLKG) -

Console-initiated strobe to the passive scan controls at CRA. (dest CRAA)

2.5.3.12 DIAG SH ACTIVE EBOX (source CLKG) -

When this signal is high, the scan path is in serial shift mode; when low, the scan path is in parallel load mode. (dest SCAH, CRAH, SHLM, SHRM, MVLF, MVRF, ESEI)

2.5.3.13 DIAG SH PAS A (source IDRM) -

When high, the passive scan paths of the CPU are in serial shift mode. When low, the passive scan paths are in parallel load mode. (dest CRAA, IDRM, MID1)

2.5.3.14 DIS PAGE FAIL (source SCAH) -

SPFN reg bit gating of page fail interrupt during EBOX memory references. (dest CRAF)

2.5.3.15 DISP Ø9, NASW=27 (i (SCAK) -

EBOX microcode dispatch to next address. Function of Y REG ØØ, Ø6:17 < or = to zero. (dest CRAG)

2.5.3.16 E PF (source MID3) -

Page fail indication for EBOX memory reference. (dest CRAG)

2.5.3.17 EA BUF Ø6-17=Ø (source CFL) -

Bits Ø6:17 = zero of the EA BUF addressed by the current EBOX SELI. Used as a branch input to EBOX microcode next address.(dest CRAG)

2.5.3.18 EARLY DCD ØØ, Ø2, Ø4, Ø5, Ø6 (source ISQ) -

Information decoded off the instruction code at instruction fetch time by the IBOX, is addressed by the EBOX SELI during EBOX execution, and used as branch inputs to the EBOX microcode next address. (dest CRAG)

2.5.3.19 EBOX AC Ø9:12 (source CFL) -

The AC field of the instruction currently in execution by the EBOX. (dest SCAH, MIDC, CRAG)

2.5.3.20 EBOX EACALC (source CFL) -

Causes an IBOX trap to the EBOX microcode in the 1st cycle of a new EBOX instruction. Also is used as a branch input to EBOX microcode, indicating that the IBOX was unable to perform the correct effective address calculation. (dest CRAG)

2.5.3.21 EBOX STEP A (source CFL Via CRAI) -

EBOX STEP from the IBOX indicates to the EBOX that the 1st cycle in progress is a valid operation, that all information needed to begin execution of the instruction was prefetched and presented properly to the EBOX by the IBOX. At CRAH it is used to gate the clock of the GLOBAL and PC SECT = Ø indicators. At SCAJ it is used to gate the update of the FLAG HISTORY BUFF address. (dest SCAJ)

2.5.3.22 EBOX1 ACT DATA IN (source CLKG) -

Serial data input from the console to the EBOX1 active scan path. (dest ESEG)

2.5.3.23 EBOX2 ACT DATA IN (source CLKG) -

Serial data input from the console to the EBOX2 active scan path. (dest CRAI)

2.5.3.24 ESE1 INSN CODE Ø:8, PØ-8 (source ESE1) -

Instruction code of the next instruction to be executed in the EBOX. During last cycle it addresses the ESE fast rams for access of the 1st cycle's microword. Also during last cycle it is latched and held at CRA for access of the 2nd microcycle's control word. (dest CRA1)

2.5.3.25 EXTEND INSN (source SCAH) -

SPFN indication to microcode next address bit 2, causing entry into microcode address space reserved for the extended instruction set. (dest ESE5)

2.5.3.26 F BUS ØØ,-Ø1,Ø2,Ø3,Ø7,Ø8,1Ø,11,12 (source MVE8,9) -

F BUS is the 38 bit result of the main ALU. It is clocked at a delayed phase 2. Some of the F BUS high-order bits are sent to CRA as Next Address dispatch inputs. F BUS -Ø1 and F BUS -Ø2 are latched at CRAI with a Clock 1 and sent to SHLH D MUX inputs as F REG -Ø1 and F REG -Ø2. F BUS ØØ and F BUS -Ø1 are sent to SCAE for generation of the overflow indication. (dest CRAI, CRAG, SCAE)

2.5.3.27 FLAGS TO # FLD (source SCAE) -

Decode of flags field returned to ESE from SCA to cause the SCA flags MUX to be selected into the next microcycle's number field. See EBOX MICROCODE CONTROL SPECIFICATION under 'FLAG CONTROL'. (dest ESEI)

2.5.3.28 FLAGS ØØ:17, 2P (source SCAF) -

Output of FLAGS MUX from SCA to the next microcycle's number field. See EBOX MICROCODE CONTROL SPECIFICATION under 'FLAG CONTROL'. (dest ESEE)

2.5.3.29 FORCE 1ST TO CRA, ESE (source IMXG) -

Signal from IBOX that indicates that the IBOX has not completed the prefetch of information necessary for the EBOX to execute the instruction. Causes ESE and CRA to hold in 1st cycle, while blocking destructive stores that may be issuing from the EBOX control store. (dest CRAH, ESEK)

2.5.3.3Ø FPA RESPONSE (source FPA) -

Input to EBOX microcode next address branch from FPA.(dest CRAG)

2.5.3.31 GLOBAL A (source CFL) -

This indicator is set by the initial EACALC that the IBOX performs on an instruction. It may be changed under microcode control between the time of the 1st EACALC and EBOX execution. It is selected into the GLOBAL signal to MVL for operation in carry control logic of the main ALU. See EBOX MICROCODE CONTROL SPECIFICATION under 'CARRY CONTROL'. (dest CRAH)

2.5.3.32 HOFF UW CLOCKS (source FPA) -

A signal from the FPA which causes the EBOX microcode to hold in a given microcycle, for use in synchronizing the FPA result on the A BUS to the EBOX acceptance of it. (dest ESEI)

2.5.3.33 HOLDOFF UW CLOCKS (source ESEI) -

A signal from the FPA which causes the EBOX microcode to hold in a given microcycle, for use in synchronizing the FPA result on the A BUS to the EBOX acceptance of it. (dest CRAH)

2.5.3.34 I,OP2,OP3 PF (source MID3) -

A trap to the EBOX microcode at 1st cycle, and branch input to EBOX microcode next address, indicating that a page fail has occurred on an IBOX prefetch of instruction or data for the current SELI. (dest CRAG)

2.5.3.35 IBOX RESPONSE (source ISQ) -

Branch input to EBOX next address, from IBOX microcode special function. (dest CRAG)

2.5.3.36 INDIRECT (source MID2) -

A trap to EBOX microcode at 1st cycle, and a subsequent branch input to EBOX microcode next address, indicating that the instruction's indirect bit (13) is on. (dest CRAG)

2.5.3.37 INSN CODE Ø:8, PØ-8 (source IDLE) -

Instruction code of the next instruction to be executed in the EBOX. During last cycle it addresses the ESE fast rams for access of the 1st cycle's microword. Also during last cycle it is latched and held at CRA for access of the 2nd microcycle's control word. (dest ESE1)

2.5.3.38 INT EBOX (source CRAK) -

Active when an EBOX trap is in progress, while the trap address is being gated to the EBOX microcode address. Its purpose is to block any destructive stores from ESE or SCA until the trap handling routine is fully entered. (dest ESEK)

2.5.3.39 INT TRAP ON (source SCAA) -

When turned off, prevents interrupts from occurring to the EBOX microcode. (dest CRAF)

2.5.3.40 INTERRUPT REQ (source SCAA) -

Interrupt to the EBOX microcode from Software PI, IO PORTS, the APR system, or Timers. (dest CRAF)

2.5.3.41 INTER UW 78 (source CRAC) -

(dest ESEK)

2.5.3.42 INTR DIS Ø:2 (source SCAA) -

The level of the current INTERRUPT REQ from Software PI, IO PORTS, the APR system, or Timers. (dest CRAG)

2.5.3.43 IO BUSY -

2.5.3.44 J MUX > 44 (source SCA3) -

Branch bit (dispatch) to EBOX next address indicating that J MUX contains a value greater that the shift capability of the Shift Matrix at SHL, SHR. (dest CRAG)

2.5.3.45 L EARLY EVEN -

2.5.3.46 LAST CY TO CRA -

2.5.3.47 L2MQ REG 32:35 -

2.5.3.48 MASK TO 44 (source SCA9) -

Branch bit (dispatch) to EBOX microcode next address indicating that the current S BUS value was forced to 44 (shift of 36 positions) due to its selected input, SC REG or SC ALU, having a value greater that 44. (dest CRAG)

2.5.3.49 MGRANT TO EBOX -

2.5.3.50 NEXT ADR ØØ:11,P -

2.5.3.51 OP2 AC N=N+1 -

2.5.3.52 OP2 GLB -

2.5.3.53 OP2=AC, OP2=AC A -

2.5.3.54 PAS DATA IN -

2.5.3.55 PC SECT=ZERO -

2.5.3.56 PCU/USER IO FLAG (source SCAG) -

PCU/USER IO FLAG goes to CRA for next address branching: to MID for control of previous context memory access and determination of VMA Ø5. (dest CRAI, MIDC)

2.5.3.57 PHASE Ø:3 TO CRA -

2.5.3.58 PHASE Ø:3 TO ESE -

2.5.3.59 PREV EN -

2.5.3.60 RESET CRA ERROR (source CLKG) -

Console controlled clear of error latches on CRA and SCA modules. (dest SCA2, CRA7)

2.5.3.61 RESET ESE -

2.5.3.62 RESET ESE ERROR -

2.5.3.63 SC REG OVFL (source SCA5) -

Input to EBOX microcode next address branch (dispatch) indicating carry out of bit 00 of the SC ALU, latched into the SC REG. (dest CRAG)

2.5.3.64 SC REG 00,02,03 (source SCA9) -

SC REG bits to EBOX microcode next address branch dispatch. (dest CRAG)

2.5.3.65 SC REG = 0 (source SCA9) -

Contents of SC REG = 0, to EBOX next address dispatch. (dest CRAG)

2.5.3.66 SCAD ALU = 0 (source SCA8) -

Contents of SC ALU = 0, to EBOX next address dispatch. (dest CRAG)

2.5.3.67 SELECT SLOW UM -

2.5.3.68 SLOW RAM 64 -

2.5.3.69 SP FCN Ø:1 UW 25:26 -

2.5.3.70 TRAP 1/2 REQ (source SCAG) -

TRAP1 FLAG or TRAP2 FLAG was set during the previous instruction; an interrupt request to EBOX microcode at 1st cycle. (dest CRAF)

2.5.3.71 USER FLAG (source SCAG) -

USER FLAG to EBOX next address dispatch at CRA, and to previous context memory access selection and VMA Ø5 at MID. (dest CRAI, MIDC)

2.5.3.72 UW Ø7 IN -

2.5.3.73 WRITE UW -

2.5.3.74 X REG 18, 28:35 (source SHR6) -

X Register bits from SHR. Bit 18 is used as a dispatch to CRA next address, as an input to the J MUX at SCA3, and as an input to the Shift Matrix at SHL. Bits 28:35 are used as inputs to the J MUX at SCA3, and the Shift Matrix at SHL. (dest CRAG, SCA1, SHL4)

2.5.3.75 X REG ØØ -

2.5.3.76 XR INVALID -

2.5.3.77 X1=X MUX -

2.5.3.78 Y REG ØØ:17 (source SHL6) -

Y Register bits from SHL. Bits Ø:3,13 are used as dispatches to the CRA next address; bits ØØ:17 are used as inputs to the Shift Matrix at SHR, and as inputs to K MUX and flags at SCA. (dest CRAG, CRAH, SHR4, SCA1)

2.5.3.79 Y REG Ø2:Ø5=Ø, 14:17=Ø (source SCAK) -

EBOX microcode next address branch inputs indicating the status of Y REG bits. (dest CRAG)

2.5.3.8Ø ØØ-Ø8, Ø9-17, 18-26, 27-35 EQUAL (source MVEM) -

The compare result of F BUS to -1, +1, or Ø according to the setting of the EBOX microcode COMP field. These signals go to CRA as Next Address dispatch inputs. (dest CRAI)

2.5.3.81 1MSEC REQ (source SCAB) -

1 milli-second interval interrupt to EBOX microcode requesting that the Time Base Counter be read and then cleared. (dest CRAF)

2.5.4 Interface: Outputs From ESE, CRA Modules

2.5.4.1 ABORT FPA -

2.5.4.2 AC=Ø (source CRAI) -

AC field of the EBOX instruction is zero. Causes block of write to AC if STORE FIELD 3. Also will block N+1 forwarding compare at CFL for STORE FIELD 3. See EBOX MICROCODE CONTROL SPECIFICATION under "STORE CONTROL". (dest ESEJ)

2.5.4.3 ALU ØØ-17 = Ø (source CRAI) -

Signal which is the "and" at CRA of F BUS ØØ:Ø8 = zero and F BUS Ø9:17 = zero; causes the set of TRAP 2 flag if FLAG field decode 14. (dest SCAG)

2.5.4.4 ALU Ø:3 UW ØØ:Ø3 (source ESEB, ESEC) -

Timing: CLK Ø gated by previous machine cycle = ODD CYCLE. ALU control field from EBOX microcode. See EBOX MICROCODE CONTROL SPECIFICATION under ALU. (dest MVEL, MVEM)

2.5.4.5 BLOCK STEP A -

2.5.4.6 CARRY OUT OF ALU (source CRAI) -

Latched at clock 1 on CRA, is the carry out of bit Ø of the previous cycle's main ALU operation at MVL/MVR. At SHRI it is used as an input to the L2MQ register, and will be shifted into L2MQ REG 35 at the following phase 3 if an L2MQ shift left of 1 is specified by the D FIELD of the EBOX microcode.

2.5.4.7 CC Ø:2 UW Ø4:Ø6 (source ESEB,C) -

Timing: CLK Ø gated by previous machine cycle = ODD CYCLE. Carry Control field from EBOX microcode, determining the mode of operation of carry into bit 17 and carry into bit 35 of the main ALU. See EBOX MICROCODE CONTROL SPECIFICATION under CARRY CONTROL. (dest MVEI)

2.5.4.8 CMPR TO 7S UW 17, CMPR TO 1 UW 18 (source ESED) -

Timing: CLK 1 gated by previous machine cycle = ODD CYCLE. Comp field from EBOX microcode, used for compare to zero, minus one, or plus one, of the result of a masking operation of SEL OP1 and SEL OP2. See EBOX MICROCODE CONTROL SPECIFICATION under COMP. (dest MVEM)

2.5.4.9 CRA ERR -

2.5.4.10 CRY OUT -

2.5.4.11 CTRL ENC VALID ECL -

2.5.4.12 D Ø:2 UW 1ØØ:1Ø2 (source CRAC) -

Controls shifting/hold of L2MQ and D MUX input selection. See EBOX MICROCODE CONTROL SPECIFICATION under D FIELD. (dest SHLJ, SHRJ)

2.5.4.13 DIAG ACT OUT CRA (source CRA7) -

Serial output of EBOX2 diagnostic scan path from CRA. (dest SCA9)

2.5.4.14 DIAG ACT OUT ESE (source For MVR=ESEJ) -

Serial data input for the MVL,MVR portions of the EBOX1 active scan path. (dest MVLF,MVRF)

2.5.4.15 DIAG CLK UW ADR (source CRAA) -

Console strobed clock to propagate the RETURN ADDRESS on CRA to the next address paths of CRA and ESE. (dest ESEA)

2.5.4.16 DIAG PAS DATA CRA -

2.5.4.17 EBOX REQ TO MEM -

2.5.4.18 EBOX RESP -

2.5.4.19 EBOX RESP FPA -

2.5.4.2Ø EBOX STEP B (source CFL Via CRAI) -

EBOX STEP from the IBOX indicates to the EBOX that the 1st cycle in progress is a valid operation, that all information needed to begin execution of the instruction was prefetched and presented properly to the EBOX by the IBOX. At CRAH it is used to gate the clock of the GLOBAL and PC SECT = Ø indicators. At SCAJ it is used to gate the update of the FLAG HISTORY BUFF address. (dest SCAJ)

2.5.4.21 EN AC WRITE A,B (source ESEJ) -

Write control for the current AC sets at SHL and SHR. When active, a write to the ACs will occur in the following cycle. (dest SHL8, SHR8)

2.5.4.22 EN MAS AC (source ESEJ) -

Write control for ths Master AC sets at SHL and SHR. When active, a write to the Master ACs will occur in the following cycle. (dest SHLD, SHRD)

2.5.4.23 ESE1 INSN CODE 0:8, P0-8 (source ESE1) -

Instruction code of the next instruction to be executed in the EBOX. During last cycle it addresses the ESE fast rams for access of the 1st cycle's microword. Also during last cycle it is latched and held at CRA for access of the 2nd microcycle's control word. (dest CRA1)

2.5.4.24 F REG -1, -2 -

2.5.4.25 FE 0:1 UW 68:69 (source CRA7) -

Control clocking of the FE MUX. See EBOX MICROCODE CONTROL SPECIFICATION under FE FIELD. (dest SCA7)

2.5.4.26 FE CTL (source ESED) -

Controls clocking of the FE MUX at SCA6; from EBOX ucode bit 17. FE will clock in EVEN CYCLE if bit 17 on, in ODD CYCLE if bit 17 off. See EBOX MICROCODE CONTROL SPECIFICATION under 'COMP'. (dest SCA7)

2.5.4.27 FLAG 0:3 UW 29:32 (source ESED, CRAD) -

The FLAG field determines the update of flags and counters resident on SCA. see EBOX MICROCODE CONTROL SPECIFICATION under 'FLAG CONTROL'. (dest SCAE)

2.5.4.28 FORCE ALU ERR (source CRAA) -

This signal is strobed on by diagnostic controls at CRA. MVL and MVR ALU err latches will be set by turning FORCE ALU ERR on and leaving it on while causing CLK 1 GATED A (MVLM, MVRM) to occur. (dest MVLL, MVRL)

2.5.4.29 FORCE 1ST TO SCA (source CFL Via CRAH) -

Force 1st will prevent the update of flags on SCA, and control the the setting of ODD and EVEN CYCLE latches at ESE and CRA. Force 1st indicates that the instruction is not valid, that the IBOX not yet completed the prefetch of information necessary for the EBOX to execute the current instruction. (dest SCA7)

2.5.4.30 GLOBAL (source CRAH) -

Under CARRY CONTROL, GLOBAL may be used to determine whether The main ALU will be in 36 or 18 bit mode. If GLOBAL is on, it may be selected to inhibit a carry into bit 17 of the ALU. (dest MVLI)

2.5.4.31 HOLDOFF UW CLOCKS (source ESEI) -

A signal from the FPA which causes the EBOX microcode to hold in a given microcycle, for use in synchronizing the FPA result on the A BUS to the EBOX acceptance of it. (dest CRAH)

2.5.4.32 INT EBOX (source CRAK) -

Active when an EBOX trap is in progress, while the trap address is being gated to the EBOX microcode address. Its purpose is to block any destructive stores from ESE or SCA until the trap handling routine is fully entered. (dest ESEK)

2.5.4.33 INTER UW 78 -

2.5.4.34 J 0:2 UW 71:73 (source CRAC) -

EBOX microcode control of the J MUX selects. See EBOX MICROCODE CONTROL SPECIFICATION under "J FIELD". (dest SCA3)

2.5.4.35 K Ø:2 UW 74:76 (source CRAC) -

EBOX microcode control of the K MUX selects. See EBOX MICROCODE CONTROL SPECIFICATION under "K FIELD". (dest SCA4)

2.5.4.36 L EARLY B (source CRA7) -

Gates the clocking of F BUS and RF BUS latches. In a 44ns microcycle, under L EARLY EVEN and L EARLY ODD control from EBOX microcode, F and RF may be clocked in the even cycle (first half), odd cycle (second half), or both even cycle and odd cycle.

2.5.4.37 L EARLY EVEN -

2.5.4.38 L EARLY ON (source CRA7) -

Gates the clocking of the FLAGS under the control of the FLAG FIELD. L EARLY ON will enable flags at SCAB and SCAC to set at clock 2 of the next machine cycle. (dest SCAB)

2.5.4.39 L EARLY ON A,B (source CRA7) -

Controls the clocking of Z MUX, X REG, and Y REG on SHL and SHR. If the source for Z MUX is the shift matrix, L EARLY on will block the clocking of Z at clock 3 of the next machine cycle. If the source for Z MUX is the MAC set, L EARLY has no effect on the clocking of Z. Step AC Adr Dlyd 2, Last Cycle, or "not" L EARLY ON enables the clocking of X and Y REGS.(dest SHLH, SHRH)

2.5.4.4Ø L UW 23 (source ESED) -

Controls the selection of RF BUS or F BUS through the L MUX. See EBOX MICROCODE CONTROL SPECIFICATION under L. (dest MVEL)

2.5.4.41 LAST CY TO CRA -

2.5.4.42 LAST CYCLE -

2.5.4.43 LAST CYCLE A (source ESEK) -

From EBOX microcode LAST CYCLE control, causes ODD CYCLE at SCA to synchronize with ODD CYCLE at ESE and CRA. (dest SCA7)

2.5.4.44 LAST CYCLE T1 C,D (source CRAI) -

Last cycle control from microcode causes clocking of Y AC read address for the next instruction. (dest SHRF,SHLF)

2.5.4.45 LD PXCT -

2.5.4.46 LD SPFN REG (source CRAF) -

Control to load the SPFN REG from the # FIELD. The SPFN REG is cleared at CLK 3 of Last Cycle. LD SPFN REG also gates # FIELD t-o enable other special functions not contained in the SPFN REG, such as FORCE PHYSICAL. See EBOX MICROCODE CONTROL SPECIFICATION under 'SPFN'. (dest SCAH)

2.5.4.47 LSW Ø:1 Uw Ø9:1Ø (source ESEB, ESEC) -

Control the selection of bits ØØ:17 and associated parity of the SW MUX (also called PASS OP). See EBOX MICROCODE CONTROL SPECIFICATION under SW/ X REG RD. (dest MVE1-5)

2.5.4.48 MAC TO Z REG (source CRAF) -

EBOX microcode SPFN 3 causes selection of the shift matrix to the Z MUX via 'not' MAC TO Z REG. At all other times MAC is selected to Z. (dest SHR3, SHL3)

2.5.4.49 MARK SYNC TP -

2.5.4.50 MQ UW 103 (source CRAD) -

Controls the mode of operation of the L2MQ REG. If off, the L2MQ performs a parallel load with data from the L1 REG. If on, the L2MQ shifts per D MUX control (if the D MUX performs a non shift operation, the L2MQ holds it's value.(dest SHRJ, SHLJ)

2.5.4.51 NEXT ADR 00:11 -

2.5.4.52 OP1=1S (source CRAF) -

A signal resulting from an EBOX special function field micro-order, causes a set of SEL OP1 data bits and a reset of SEL OP1 parity bits. It allows SEL OP2 to pass through to the COMP BUS and-gates for compare to 0, -1, or +1. See EBOX MICROCODE CONTROL SPECIFICATION under SP FUNCTION. (dest MVE1)

2.5.4.53 OP1 UW 07 (source ESEB, ESEC) -

From EBOX microcode, controls selection into the SEL OP1 MUX of either AF MUX direct or AF MUX with halfwords swapped. See EBOX MICROCODE CONTROL SPECIFICATION under OP1/WR REG FI. (dest MVE1-4)

2.5.4.54 OP2=1S (source CRAF) -

A signal resulting from a EBOX special function field micro-order, causes a set of SEL OP2 data bits and a reset of SEL OP2 parity bits. It allows SEL OP1 to pass through to the COMP BUS and-gates for compare to 0, -1, or +1. See EBOX MICROCODE CONTROL SPECIFICATION under SP FUNCTION (dest MVEH)

2.5.4.55 PREV EN A (source ESEJ) -

Control to select the previous AC block number to address the MAC and AMC backup. (dest SCAH)(dest SCAH)

2.5.4.56 R 0:3 UW 13:16 (source ESEB, ESEC) -

From EBOX microcode, controls the selection of data inputs to the R MUX. See EBOX MICROCODE CONTROL SPECIFICATION under R/ V REG RD. (dest MVE6)

2.5.4.57 RF UW 104 (source CRAD) -

From EBOX microcode, controls selection of RF MUX to be R BUS if off, or A BUS, if on. See EBOX MICROCODE CONTROL SPECIFICATION under RF. (dest MVEL)

2.5.4.58 REG WR UW 07 (source ESEB,C) -

Controls the write of the X and Y REG files in conjunction with L EARLY. See EBOX MICROCODE CONTROL SPECIFICATION under OP1/WR REG FILE. (dest SHR7,SHL7)

2.5.4.59 RSW 0:1 UW 11:12 (source ESEB, ESEC) -

Control the selection of bits 18:35 and associated parity of the SW MUX (also called PASS OP). See EBOX MICROCODE CONTROL SPECIFICATION under SW/ X REG RD. (dest MVE1-5)

2.5.4.60 SC CTL (source ESED) -

From COMP field bit 17. If Uw 17 is on, SC REG will clock in EVEN CYCLE; if UW 17 is off SC REG will clock in ODD CYCLE. See EBOX MICROCODE CONTROL SPECIFICATION under "COMP". (dest SCA7)

2.5.4.61 SC 0:2 UW 65:67 (source CRAC) -

EBOX microcode control of the SC MUX selection and the clocking of the SC REG. See EBOX MICROCODE CONTROL SPECIFICATION under "SC FIELD". (dest SCA7)

2.5.4.62 SCU ALU UW 77 (source CRAB) -

EBOX microcode control which determines the mode of operation of the SC ALU. If UW 77 is on SC ALU is in subtract mode; if UW 77 is off SC ALU is in add mode. (dest SCA7)

2.5.4.63 SEL AC ADR ALU -

2.5.4.64 SEL #FLD (i (source CRAE) -

From UW bit 36. If on, cause selection of # FIELD 10:13 to the MAC and MAC BACKUP hi-order 4 address bits from SCA. UW 36 on at SHR and SHL will cause selection of # FIELD 14:17 to MAC and MAC BACKUP lo-order 4 address bits. (dest SCAH)

2.5.4.65 SELECT SLOW UW -

2.5.4.66 SLOW RAM 64 -

2.5.4.67 SP FCN 0:1 UW 25:26 -

2.5.4.68 SPFN ICMD -

2.5.4.69 SPFN LD BLK (source CRAF) -

EBOX microcode SPFN decode 2 to load the current and previous AC blocks. Current from Y REG 0:2; Previous from Y REG 3:5. (dest SCAH)

2.5.4.70 SPFN LD IO REG (source CRAF) -

EBOX microcode SPFN decode 6 to load IO REG with control information for the IO PORTS. A strobe from CRA signals to the IOP that the IO REGISTER signals are valid. See EBOX MICROCODE CONTROL SPECIFICATION under "SPFN". (dest IOP)

2.5.4.71 SPFN LD PI (source CRAF) -

EBOX microcode SPFN decode 1 to load the PI system with control infromation. See EBOX MICROCODE CONTROL SPECIFICATION under "SPFN". (dest SCAA)

2.5.4.72 SPFN STEP ACCT -

2.5.4.73 STEP AC ADR (source ESEJ) -

Caused by STORE CONTROL field decode 7, this signal is used to step AC addresses while a block of ACs is being loaded from the MAC set to the current AC set. (dest SHRJ, SHLJ)

2.5.4.74 STOP CLOCK -

2.5.4.75 T UW 78 -

2.5.4.76 UW Ø7 IN -

2.5.4.77 WR AC TO FWD -

2.5.4.78 WR Ø:3 UW 33:36 (source ESED, CRAE) -

Write address for the X and Y REG files. See EBOX MICROCODE CONTROL SPECIFICATION UNDER "WR".(dest SHRF, SHLF)

2.5.4.79 WRITE BKUP (source ESEJ) -

Control signal caused by EBOX microcode STORE field will enable a write of L REG data into the MAC backup RAMs. (dest MVEF)

2.5.4.8Ø WRITE MEM TO IBOX -

2.5.4.81 WRITE UW -

2.5.4.82 X Ø:2 UW 41:43 (source CRAC) -

EBOX microcode control of the X MUX selects. See EBOX MICROCODE CONTROL SPECIFICATION under "X". (dest SHR6,J, SHL6,J)

2.5.4.83 XRD Ø:3 UW Ø9:12 (source ESEB,ESEC) -

Read address of the X REG file. See EBOX MICROCODE CONTROL SPECIFICATION under "SW/XRD". (dest SHRA, SHLA)

2.5.4.84 X1 Ø:1 UW 37:38 (source ESEK) -

Control field from EBOX microcode which determines the selection of inputs to the X1 MUX. See EBOX MICROCODE CONTROL SPECIFICATION under X1. (dest MVE1-5)

2.5.4.85 Y Ø:1 UW 44:45 (source CRAC) -

EBOX microcode control of the Y MUX selects. See EBOX MICROCODE CONTROL SPECIFICATION under "Y". (dest SHR9,SHL9)

2.5.4.86 YRD Ø:3 UW 13:16 (source ESEB,ESEC) -

Read address of the Y REG file. See EBOX MICROCODE CONTROL SPECIFICATION under "R/YRD" .(dest SHRA, SHLA)

2.5.4.87 Y1 Ø:1 UW 39:4Ø (source ESEK) -

Control field from EBOX microcode which determines the selection of inputs to the Y1 and AF MUXes. See EBOX MICROCODE CONTROL SPECIFICATION under Y1. (dest MVE 1-5,L)

2.5.4.88 # FLD ØØ:17 UW 8Ø:97 (source ESEE) -

Input to J MUX, K MUX, flags on SCA. # FLD bits ØØ:13 go to MID, latched by SPFN ICMD, as qualifiers for memory operations. # FLD bits 14:17 go to ISQ as the 4 lo-order bits of the IPUT microcode address of the 1st IPUT microword of an EBOX routine. See EBOX MICROCODE CONTROL SPECIFICATION under "# FLD" and "SPFN ICMD". (dest SCA2, MID8, ISQH)

2.5.4.89 # FLD 00:17, 2P (source ESEE,ESEF) -

EBOX microcode input to the data path via the X MUX. See EBOX MICROCODE CONTROL SPECIFICATION under "#". (dest SHR6, SHL6)

2.6 EBOX MICROCODE CONTROL SPECIFICATION

The JUPITER EBOX control storage is comprised of 4K words of 106 bits and 512 words of 30 bits each. The 512X30 bit array overlays the 4KX106 array in such a way as to allow 30 active bits and 76 default bits in 1st cycle, and 106 active bits in 2nd and successive cycles of the instruction. A wide word permits coding flexibility without extending control storage depth. The fields are defined as follows:

UWORD BIT#	FIELD	FLD#	CARD	FAST RAM/SLOW RAM,TIMING
0:3	ALU	0:3	ESE	F/S,E CLK 0
4:6	CC	0:2	ESE	F/S,E CLK 0
7	OP1/REG WR		ESE	F/S,E CLK 0
8	LAST CYCLE		ESE	F/S,E CLK 0
9:12	SW/XRD	0:3	ESE	F/S,I CLK 1
13:16	R/YRD	0:3	ESE	F/S,I CLK 1
17:18	COMP	0:1	ESE	F/S,I CLK 1
19:21	STORE	0:2	ESE	F/S,I CLK 1
22	P ESE		ESE	F/S,I CLK 1
23	L		ESE	F/S,L CLK 3
24	MARK		ESE	F/S,L CLK 3
25:26	SPFN	0:1	ESE	F/S,E CLK 0
27:28	SPFN	2:3	CRA	S,E CLK 0
29:30	FLAG	0:1	ESE	F/S,I CLK 1
31:32	FLAG	2:3	CRA	S,I CLK 1
33	WR	0	ESE	F/S,E CLK 0
34:36	WR	1:3	CRA	S,I CLK 1
37:38	X1	0:1	ESE	S,E CLK 0
39:40	Y1	2:3	ESE	S,E CLK 0
41:43	X	0:2	CRA	S,E CLK 0
44:45	Y	0:1	CRA	S,E CLK 0
46:57	NAF	0:11	CRA	S,E CLK 0
58:63	NASW	0:5	CRA	S,E CLK 0
64	L EARLY EVEN		CRA	S,E CLK 0
65:67	SC	0:2	CRA	S,I CLK 1
68:69	FE	0:1	CRA	S,I CLK 1
70	CALL		CRA	S,I CLK 1
71:73	J	0:2	CRA	S,I CLK 1
74:76	K	0:2	CRA	S,I CLK 1
77	SC ALU		CRA	S,I CLK 1
78	T		CRA	S,I CLK 1
79	P CRA		CRA	S,I CLK 1
80:97	#	0:17	ESE	S,L CLK 3
98	#PAR0		ESE	S,L CLK 3
99	#PAR1		ESE	S,L CLK 3
100:102	D	0:2	CRA	S,I,L CLK 1(MQ ctrl),CLK 3(D ctrl)
103	MQ		CRA	S,I CLK 1
104	RF		CRA	S,L CLK 3
105	L EARLY ODD		CRA	S,L CLK 0

2.6.1 ALU (UW Bits Ø:3)

The ALU field controls the 36 bit ALU on the MVL and MVR modules. Carry into 35 must be asserted for all subtract operations. For BCD, bits Ø,9,18,27 are ignored <carry will cross from 1Ø to 8, 19 to 17, 28 to 26> during the ALU operation and forced to zero at the F BUS. The decodes are as follows:

decode	function
Ø	X1 plus Y1 BCD (32 bit add, 1-8,1Ø-17,19-26,28-35)
1	X1 equal Y1
2	X1 plus Y1 binary (36 bit)
3	not Y1
4	Y1 minus X1 BCD (32 bit subtract)
5	X1 + Y1(inclusive or)
6	Y1 minus X1 Binary (36 bit)
7	X1.Y1(logical and)
1Ø	X1 minus Y1 BCD (32 bit subtract)
11	X1 EOR Y1
12	invalid op
13	Y1
14	Ø minus Y1 BCD (32 bit subtract)
15	X1
16	invalid op
17	zeros

2.6.2 CARRY CONTROL (UW Bits 4:6)

The CC field controls the carry into 17 and carry into 35 of the 36 binary ALU and 32 bit decimal ALU. Decodes are as follows:

DECIMAL OPERATION (UW bits 2,3=ØØ)

decode	function
Ø-1	invalid op
2	carry into 35
3-5	invalid op
6	no carry into 35
7	invalid op

BINARY OPERATION (UW bits 2,3 not equal ØØ)

decode	function
Ø	sign Y= Cn to 35, 36 bit ALU
1	no Cn to 35, 36 bit ALU
2	no Cn to 35, 18 bit ALUs
3	no Cn to 35, 18 bit ALUs if EACALC not global no Cn to 35, 36 bit ALU if EACALC global
4	Cn to 35, force Cn to 17
5	Cn to 35, 36 bit ALU
6	Cn to 35, 18 bit ALUs
7	Cn to 35, force Cn to 17 if EACALC not global Cn to 35, 36 bit ALU if EACALC global

2.6.3 OP1/WR REG FILE (UW Bit 7)

During first and subsequent cycles UW bit 7 controls the OP1 MUX at MVL and MVR as follows:

UW bit 7=0	AF MUX 0:35 to SEL OP1 0:35
UW bit 7=1	AF MUX 0:17 to SEL OP1 18:35,AF MUX 18:35 to SEL OP1 0:17

During second and subsequent cycles, UW bit 7 performs the following operation in conjunction with the L EARLY control:

UW Bit 7	L EARLY	function
1	UW 105	write REG FILE in 2nd half next cycle,
1	UW 64	write REG FILE in 1st half next cycle,

2.6.4 LAST CYCLE (UW Bit 8)*

The LAST CYCLE bit controls the update of SELI, and selection of CE, CI, and CL MUXes during the first cycle of the next instruction. When the last cycle bit is on and L EARLY is also on, SELI is updated in the next 22ns clock period (if in 44ns microcycle,during the 2nd half of the cycle containing the last cycle bit. When the LAST CYCLE bit is on and L EARLY is off, SELI is updated in the 1st 22ns period following this 44ns microcycle.

CE, CI, and CL MUXes select between first and subsequent cycle control words for early (CLK 0), intermediate (CLK 1 or 2) and late (CLK 3 or 0) control bits. LAST CYCLE bit on causes fast 256X4 RAMS and forwarding controls to be selected into the control word of the next cycle, which represents the first cycle of the next instruction.

An interrupt may block the operation of the LAST CYCLE bit, thus allowing slow microcode to trap to an interrupt handling routine. Once this routine is completed, a LAST CYCLE microorder must again be issued to cause instruction execution to continue.

Instruction invalid coming from the IBOX at LAST CYCLE will cause the microword containing the LAST CYCLE order to hold if no interrupts are pending. L EARLY will be forced off in the next cycle so that only one store operation will occur.

2.6.5 SW/ X REG RD (UW Bits 9:12)

The SW/ X REG RD field performs a dual function. It addresses the X REG FILE words 0-15 on SHL and SHR modules, and controls the swapping and merge operation of the SW MUX on MVL and MVR modules. SW MUX is con- trolled by the following decodes:

Left swap control (UW bits 9:10)

decode	function
0	OP1 18:35 to PASS 0:17
1	OP2 18:35 to PASS 0:17
2	OP1 0:17 to PASS 0:17
3	OP2 0:17 to PASS 0:17

Right swap control (UW bits 11:12>

decode	function
0	OP1 18:35 to PASS 18:35
1	OP2 18:35 to PASS 18:35
2	OP1 0:17 to PASS 18:35
3	OP2 0:17 to PASS 18:35

2.6.6 R/Y REG RD (UW Bits 13:16)

The R/Y REG RD field performs a dual function. It addresses the Y REG FILE words 0-15 on SHL and SHR modules, and controls the operation of the R MUX on MVL and MVR modules. R control bits are decoded as follows:

decode	function
0	zeros to R BUS 0:17, PASS 18:35 to R BUS 18:35
1	zeros to R BUS 0:35
2	PASS 0:17 to R BUS 0:17, zeros to 18:35
3	zeros to R BUS 0:35
4	SEL OP2 bit 0 to R BUS 0:17, PASS 18:35 to R BUS 18:35
5	PASS 0:35 to R BUS 0:35
6	PASS 0:17 to R BUS 0:17, SEL OP2 bit 0 to R BUS 18:35
7	COMPARE BUS 0:17 to R BUS 0:17
10	ones to R BUS 0:17, PASS 18:35 to R BUS 18:35
11	ones to R BUS 0:35
12	PASS 0:17 to R BUS 0:17, ones to R BUS 18:35
13	ones to R BUS 0:35
14	SEL OP2 bit 18 to R BUS 0:17, PASS 18:35 to R BUS 18:35
15	-PASS 0:35 to R BUS 0:35
16	PASS 0:17 to R BUS 0:17, SEL OP2 bit 18 to R BUS 18:35
17	-COMPARE 0:35 to R BUS 0:35

2.6.7 COMP (UW Bits 17:18)

The COMP field determines the mode of operation of the comparators on the MVL and MVR modules per the following decodes:

UW Bit 17 on= compare to ones(7s)
a)compare COMP BUS 1:34 to ones
b)compare F BUS Ø:35 to ones
UW Bit 17 off= compare to zeros(not 7s)
a)compare COMP BUS 1:34 to zeros
b)compare F BUS Ø:35 to zeros
UW Bit 18 on= compare to 1
a)compare COMP BUS 35 to one
UW Bit 18 off= compare to zero
a)compare COMP BUS 35 to zero

The COMP field UW bits 17 and 18 also determine the mode of clocking of the SC and FE regs in the following manner:

UW 17 on= clock SC REG early with results of the SC ALU operation occurring in the 1st 22ns period of a 44ns microcycle
UW 17 off= clock SC REG late with results of the SC ALU operation occurring in the 2nd 22ns period of a 44ns microcycle
UW 18 on= clock FE REG early with the results of the SC ALU operation occurring in the 1st 22ns period of a 44ns microcycle
UW 18 off= clock FE REG late with the results of the SC ALU operation occurring in the 2nd 22ns period of a 44ns microcycle

2.6.8 STORE CONTROL (UW Bits 19:21)

The STORE CONTROL field causes store and MBOX operations to occur per the following decodes:

decode	function
Ø	no op
1	write MEM with data on L BUS. Write AC if EACALC is an AC. L EARLY on will cause data to latch at M or write to occur at AC in the 1st half of the next cycle. L EARLY off will cause data to latch at M or write to occur at AC in the 2nd half of the next cycle.
2	Write to AC and MASTER AC. L EARLY on will cause the write to occur in the 1st half of the next cycle. L EARLY off will cause the write to occur in the 2nd half of the next cycle.
3	Write AC and MAC if AC field not=Ø. Timing is the same as decode 2.
4	Write MAC only with data on L BUS. L EARLY on will cause the write to occur in the 1st half of the next cycle. L EARLY off will cause the write to occur in the 2nd half of the next cycle.
5	spare
7	STEP CTRS: Read the MASTER AC twice followed by an update of it's lo-order 4 address by +1. In the next microcycle write into AC copies twice followed by an update of their write address by +1. This microorder repeated 8 times will cause a load of all AC copies with an AC set from the MASTER AC block.

2.6.9 PARITY ESE (UW Bit 22)

Parity ESE is turned on or off so that odd parity is achieved for 53 of the 54 microcode bits located on the ESE module (1:23,25:26,29:3Ø, 33, 37:4Ø,8Ø:99). The MARK bit is not parity checked.

2.6.1Ø L (UW Bit 23)

The L bit controls the selection of the L MUX on MVL and MVR modules according to the following decode:

L=Ø F BUS Ø:35 to L MUX Ø:35
L=1 RF BUS Ø:35 to L MUX Ø:35

2.6.11 MARK (UW Bit 24)

The MARK bit is used as a diagnostic scoping tool. It provides a sync pulse for monitoring dynamic CPU activity. The MARK bit may be changed at any time without affecting the odd parity of the ESE microcode bits.

2.6.12 SP FUNCTION Field (UW Bits 25:28)

The SPECIAL FUNCTION field issues control commands to the EBOX under the following decodes:

decode	function
0	no-op
1	LD PI: Set interrupt system on if Y REG bit 10 is on, set interrupt system off if Y REG bit 10 is off. Load software requests 7:1 from X REG bits 29:35. Load current level in progress 1:7 from Y REG 3:9. Load "levels on" 1:7 from Y REG 11:17.
2	LD BLK: Load new current AC block number from Y REG 0:2. Load previous AC block number from Y REG 3:5.
3	SEL SHFT: Select shift matrix output to Z MUX
4	force 1s to SEL OP2
5	STEP ACCT: update accounting meter +1 (not currently implemented)
6	LD IO: Load IO register from X REG 28:35. Once the IO register has been loaded, provide a 22ns IO REG STROBE to the IOBOX to enable a decode at the IOBOX
7	IBOX CMD: Command to IBOX per #field 14:17. #field 0:12 are latched and held as MBOX qualifiers during the next EBOX trap of IPUT. L BUS from this microcycle is held at L3 until the next IBOX CMD or until a write to AC with IPUT not in an EBOX routine takes place. If a write to AC with IPUT in an EBOX routine takes place, IBOX trap to EBOX is raised to cause a reload of the current AC set.

IO REG 28:29	CTRL ENC 0 no op 1 doorbell 2 console interrupt 3 console boot
IO REG 30	reset
IO REG 31	broadcast
IO REG 32:35	port select or intr level

10 force 1s to SEL OP1

11 EBOX RESP FPA: EBOX response to FPA to free a microcode interlock condition (not currently implemented)

12 LD SPFN: Load Special function register from #field 0:17. This register is cleared at last cycle.

# FLD 00	DIS PF: Disable call to microcode location 4016 on EBOX page fail
# FLD 01	ALLOW RETRY: If on, allow auto retry of this instruction. If off, block auto retry of this instruction.
# FLD 02	spare
# FLD 03	CLEAR DP...Reset EBOX and IBOX data paths for retry (must be followed by a load of current AC set from MAC) note: this reset does not reset CRA or MBOX interface buffers in IBOX. These must be reset by the console.
# FLD 04	SET FORCE PHY: Set force physical F/F causing all VMA references to be physical
# FLD 05	CLR FORCE PHY: Clear the force physical F/F
# FLD 06	CLR JREAL: Reset Jump Real at the IBOX IDL module if # FLD 06 is off
# FLD 07	BLK STEP SELI: Block stepping of SELI at next LAST CYCLE if # FLD 07 is off
# FLD 08	DIS ADR BREAK: Disable address break page fails from the MBOX if # FLD 08 is off.
# FLD 09:11	STATE REG: Set state register bits 1:3 to the complement of # FLD bits 9:11.
# FLD 12	EXTEND INSN: If # FLD 12 is off, force the first slow microword fetched after the next last cycle to be fetched from opcode+1000.
# FLD 13:14	Spare
# FLD 15	RESET SC OVL: Reset the SCAD ALU overflow bit
# FLD 16	DECR FLAG HIST: Decrement the flag history buffer address
# FLD 17	INCR FLAG HIST: Increment the flag history buffer address

13 ABORT FPA: Stop instruction in execution at FPA by trapping FPA microcode into a routine which will clear it's data paths and enter an idle loop (not currently implemented)

14 LD PXCT: Load PXCT flags from the AC address bits 09:12 at the IBOX. PXCT ENABLED will now come on if any of these bits are on. The PXCT flags will be

reset at the second occurrence of last cycle.

15 EBOX RESP: EBOX response to IBOX and (IBOX microcode will branch on it during interlock conditions).

16 HALT: Stop EBOX clocks and signal the console

17 Set EM MODE if FBUS bit 2 off. Clear EM mode if FBUS bit 2 on.

2.6.13 FLAG CONTROL (UW Bits 29:32)

The FLAG CONTROL field causes flag and miscellaneous operations to occur under control of # FIELD and data path bits.

decode	function
Ø	no op
1	TIME BASE: Reset time base if Y REG bit 2 is on. Load time base enable from Y REG bit 5. Load interval timer PIA from Y REG 15:17. Force zeros into the next microcycle's number field.
2	PC FLAGS: Set or reset selected PC flags according to # field bits. OV...set if #fld ØØ is on FOV...set if #fld Ø1 is on TRAP1...set if #fld ØØ is on; set to the value of -#fld Ø8 if #fld ØØ is off TRAP2...set to the value of -#fld Ø7 FPD...set if #fld Ø4 is on, clear if #fld Ø3 is on NO DIV...set if #fld Ø2 is on USER...clear user if #fld Ø5 is on
3	EN APR: Load APR enable bits from X REG 29:31. Load APR PI level from X REG 33:35. Force zeros into the next microcycle's number field.
4	AD FLAGS: set CRYØ if carry out of main ALU bit -Ø2. Leave CRYØ on if already on. Set OV and TRAP1 if F BUS ØØ is not equal to F BUS -Ø1. Leave OV and TRAP1 on if already on. Set CRY1 if OV condition is not equal to CRYØ. Leave CRY1 on if already on.
5	Unassigned. Force zeros into the next microcycle's number field.

6 LD FLAGS: load PC flags from Y REG Ø:12. If # field bit 5 is on protect USER and USER IO from illegal modification. If # field bit 9 is on load PCU from previous USER flag.

Y REG	flag
ØØ	OV
Ø1	CRYØ
Ø2	CRY1
Ø3	FOV
Ø4	FPD
Ø5	USER (if Y REG Ø5 on, set USER. If Y REG Ø5 off and #fld Ø5 on, preserve the state of USER. If not any of the above, clear USER)
Ø6	USER IO/PCU (Y REG Ø6 on and not USER set USER IO/PCU, Y REG Ø6 on and #fld Ø5 on set USER IO/PCU if USER IO/PCU previously on, #fld Ø9 on set USER IO/PCU if USER on; not any of the above, clear USER IO/PCU)
Ø8	INH ADDR BRK: inhibit address break will be operative during the instruction immediately following the next last cycle and will clear at the last cycle of the same instruction
Ø9	TRAP2
10	TRAP1
11	FXU
12	NO DIV

7 LD TIM: Load 12 bit timer interval from Y REG 6:17. Set timer on if Y REG Ø3 is on, timer off if Y REG Ø3 is off. Reset timer overflow and timer done flags if Y REG Ø4 is on. Clear interval timer to zeros if X REG 18 is on. Force zeros into the next microcycle's number field.

1Ø JFCL: Clear OV if OV and AC field bit 9 are on; clear CRYØ if CRYØ and AC field bit 1Ø are on; clear CRY1 if CRY1 and AC field bit 11 are on; clear FOV if FOV and AC field bit 12 are on. If any of these flags are cleared, force jump successful at the IBOX.

11 READ PC FLAGS: Read PC flags Ø:12 into the next microcycle's number field bits Ø:12. Clear bits 13:17 of the next microcycle's number field.

12 EXP: Exponent test (short or long) sets OV, FOV, TRAP1, or FXU from SC REG bits.

OV, FOV, TRAP1...set if #fld 1Ø on and SC REG ØØ on, set if #fld 1Ø off and SC REG Ø3 on
FXU...set if #fld 1Ø on and SC REG OVFL on, set if #fld 1Ø off and SC REG Ø2 on

13 LD APR: Set all APR flags if X REG 28 on, reset selected APR flags from X REG 29:31. Read the APR flags into the next microcycle's number field bits 1:3, read the CPU serial number into the next microcycle's number field bits 4:17

14 PDL OV: Set trap 2 if F BUS Ø:17 compare equals value of COMP field.

15 READ TIME BASE: Read the time base into the next microcycle's number field bits 1:17.

16 RESTORE: Load flags OV, CRYØ, CRY1, FOV, TRAP1, and TRAP2 from the flag history buffer. This buffer is addressed by the IBOX SELI HIST counter. During retry restore will load the state at which these flags were set just prior to the execution of the instruction being retried.

17 READ APR: Read APR FLAG ON into the next microcycle's number field bit Ø, read the state register into the next microcycle's bits 1:3, read TIMER DONE into the next microcycle's bit 4, read TIM OV into the next microcycle's bit 5, read the interval timer into the next microcycle's bits 6:17.

2.6.14 WR CTRL (UW Bits 33:36)

The AC/ REG WR CTRL field performs a dual function. IF UW bit 7 is on, the X and Y REG FILES will be written into in the next microcycle according to the setting of L EARLY. The X and Y REG write address is generated from bits UW bits 33:36.The AC/ REG WR field will also be used to control the addressing of the ACs at SHL, SHR, IDL, IDR, and FPA modules. In this area of control UW bits 33:36 have the following significance:

UW Bit 34=Ø — WR MUX plus # FIELD 14:17 to OP2 AC ADR (UW bit 34=Ø and Uw bit 7=Ø causes XRD MUX to clock).

Uw Bit 34=1 — EACALC 32:35 to OP2 AC ADR if LAST CYCLE, hold previously clocked contents if nof LAST CYCLE.

Decode of UW 33,35:

Ø Ø	Y AC RD ADR to WR ADR MUX
Ø 1	OP2 AC ADR to WR ADR MUX
1 Ø	VMA 32:35 to WR ADR MUX
1 1	zeros to WR ADR MUX

Decode of Mem Access Holdoff, Uw 36

Ø Ø	AC WR ADR MUX to lo-order 4 bits MAS AC ADR
Ø 1	# Field 14:17 to lo-order 4 bits MAS AC ADR
1 Ø	VMA 32:35 to lo-order 4 bits MAS AC ADR
1 1	# Field 14:17 to lo-order 4 bits MAS AC ADR

Hi-order 4 bits MAS AC ADR

UW Bit 36=Ø — zeros to hi-order 4 bits MAS AC ADR

UW Bit 36=1 — # Field 1Ø:13 to hi-order 4 bits MAS AC ADR

2.6.15 X1 (UW Bits 37:38)

The X1 field controls the X1 MUX at MVL and MVR. During the first cycle of an instruction, forwarding compare logic from TAG and SH modules determine selection. During second and subsequent cycles firmware has control with the following field decodes:

decode	function
Ø	F BUS Ø:35 to X1 BUS Ø:35, F BUS Ø to X1 BUS -2,-1
1	RF BUS Ø:35 to X1 BUS Ø:35, RF BUS Ø to X1 BUS -2,-1
2	X BUS -2:35 to X1 BUS -2:35
3	OP2 BUS Ø:35 to X1 BUS Ø:35,OP2 BUS Ø to X1 BUS -2,-1

2.6.16 Y1 (UW Bits 39:40)

The Y1 field controls the Y1 MUX and AF MUX at MVL and MVR (an enable to Y1 in first cycle comes from the IBOX). During the first cycle, forwarding compare logic from TAG and SH modules determine selection. During second and subsequent cycles firmware has control with the following decodes:

decode	function
0	F BUS 0:35 to AF,Y1 BUS 0:35
1	RF BUS 0:35 to AF,Y1 BUS 0:35
2	X BUS 0:35 to AF,Y1 0:35
3	Y BUS 0:35 to AF,Y1 0:35

ALU input -2,-1 are equal the value of Y1 BUS 0

2.6.17 X FIELD (UW BITS 41:43)

The X field controls inputs into the X BUS via the X MUX at the SHL and SHR modules. Decode 7 is forced if the selection is decode 0 and the XCT flag is on. The field decodes are as follows:

decode	function
0	X AC 0:35 to X BUS 0:35, X AC bit 0 to X BUS -2 and -1
1	XREG FILE 0:35 to X BUS -2:33,zeros to 34,35
2	XREG FILE 0:35 to X BUS -1:34,zero to 35, XREG FILE bit 0 to X BUS -2
3	XREG FILE 0:35 to X BUS 0:35, XREG FILE bit 0 to X BUS -2 and -1
4	D MUX 0:35 to X BUS 0:35, D MUX bit 0 to X BUS -2 and -1
5	XREG FILE bit 0 to X BUS -2:35
6	# FIELD 0:17 to X BUS 0:17, # FIELD 0:17 to X BUS 18:35, # FIELD 0 to X BUS -2 and -1
7	Z BUS 0:35 to X BUS 0:35, Z BUS bit 0 to X BUS -2 and -1

2.6.18 Y FIELD (UW Bits 44:45)

The Y field controls inputs into the Y BUS via the Y MUX at the SHL and SHR modules. The field decodes are as follows:

decode	function
0	Y AC 0:35 to Y BUS 0:35
1	YREG FILE 0:35 to Y BUS 0:35
2	D MUX 0:35 to Y BUS 0:35
3	MQ 0:35 to Y BUS 0:35

2.6.19 Next Address Field (bits 46:57)

In any given instruction, the first cycle control word comes from fast 256X4 RAMS addressed by the instruction code, the second cycle control word (22ns later) comes from 4KX1 RAMS also addressed by the instruction code, and third and subsequent control words come at 44ns intervals from the same 4KX1 RAMS addressed by the next address path. If control bits 58:63 are not Ø, the hi-order next address is determined by hi-order next address field bits, and the next address lo-order is determined by the inclusive or of lo-order next address field bits and the selected skip or dispatch bits.

2.6.2Ø THE NA SWITCH (BITS 58:63)

The NA SWITCH field determines the source of the NEXT ADDRESS path.

decode	function
Ø	NA FIELD Ø:11 to NA BUS Ø:11
1	SKIP PAGE FAULT: NA FIELD Ø:1Ø to NA BUS Ø:1Ø, E PF inclusive or with NA FIELD 11 to NA BUS 11
2	SKIP AC NOT EQL Ø: NA FIELD Ø:1Ø to NA BUS Ø:1Ø, instr AC ADDR not=Ø inclusive or with NA FIELD 11 to NA BUS 11
3	SKIP NOT Y Ø: NA FIELD Ø:1Ø to NA BUS Ø:1Ø, not Y REG bit Ø inclusive or with NA FIELD 11 to NA BUS 11
4	SKIP NOT X 18: NA FIELD Ø:1Ø to NA BUS Ø:1Ø, not X REG bit 18 inclusive or with NA FIELD 11 to NA BUS 11
5	SKIP NOT SCAD Ø: NA FIELD Ø:1Ø to NA BUS Ø:1Ø, not SCAD ALU bit Ø inclusive or with NA FIELD 11 to NA BUS 11
6	SKIP NOT SC REG Ø: NA FIELD Ø:1Ø to NA BUS Ø:1Ø, not SC REG bit Ø inclusive or with NA FIELD 11 to NA BUS 11
7	SKIP F Ø: NA FIELD Ø:1Ø to NA BUS Ø to 1Ø, F BUS Ø inclusive or with NA FIELD 11 to NA BUS 11 (from ALU operation of the 1st half of the current microcycle if L EARLY, from ALU operation of the 2nd half of the previous microcycle if not L EARLY)

10	SKIP ALU 18:35 NOT Ø: NA FIELD Ø:1Ø to NA BUS Ø:1Ø, ALU 18:35 not=Ø inclusive or with NA FIELD 11 to NA BUS 11 (same timing as decode 7)
11	SKIP ALU Ø:17 NOT Ø: NA FIELD Ø:1Ø to NA BUS Ø:1Ø, ALU Ø:17 not=Ø inclusive or with NA FIELD 11 to NA BUS 11 (same timing as decode 7)
12	SKIP ALU ØØ:35 NOT Ø: NA FIELD Ø:1Ø to NA BUS Ø:1Ø, ALU ØØ:35 not=Ø inclusive or with NA FIELD 11 to NA BUS 11 (same timing as decode 7)
13	SKIP PC SECT NOT Ø: NA FIELD Ø:1Ø to NA BUS Ø:1Ø, PC section not=Ø inclusive or with NA FIELD 11 to NA BUS 11
14	SKIP NOT SC REG 1: NA FIELD Ø:1Ø to NA BUS Ø:1Ø, not SC REG bit 1 inclusive or with NA FIELD 11 to NA BUS 11
15	SKIP ALU CO: NA FIELD Ø:1Ø to NA BUS Ø:1Ø, CARRY OUT of ALU inclusive or with NA FIELD 11 to NA BUS 11 (same timing as decode 7)
16	SKIP SC REG NEQ Ø: NA FIELD Ø:1Ø to NA BUS Ø:1Ø, SC REG not=Ø inclusive or with NA FIELD 11 to NA BUS 11
17	SKIP SCAD NOT EQL Ø: NA FIELD Ø:1Ø to NA BUS Ø:1Ø, SCAD ALU not=Ø inclusive or with NA FIELD 11 to NA BUS 11

20 DISP USER: NA FIELD Ø:Ø8 to NA BUS Ø:Ø8, USER MODE bit inclusive or with NA FIELD Ø9 to NA BUS Ø9, IO LEGAL (PCU/USER IO or not USER FLAG) inclusive or with NA FIELD 1Ø to NA BUS 1Ø, IO BUSY inclusive or with NA FIELD 11 to NA BUS 11

21 DISP OP2=AC.GLOBAL: NA FIELD Ø:Ø9 to NA BUS Ø:Ø9, EACALC global inclusive or with NA FIELD 1Ø to NA BUS 1Ø, OP2=AC inclusive or with NA FIELD 11 to NA BUS 11

22 DISP SC ALU: NA FIELD Ø:Ø8 to NA BUS Ø:Ø8, MASK TO 44 (SMUX > 44) inclusive or with NA FIELD Ø9 to NA BUS Ø9, not SC REG bit Ø inclusive or with NA FIELD 1Ø to NA BUS 1Ø, SC REG OVFL (SC ALU OVERFLOW) inclusive or with NA FIELD 11 to NA BUS 11

23 DISP MISC RESP: NA FIELD Ø:Ø7 to NA BUS Ø:Ø7, IBOX RESPONSE inclusive or with NA FIELD Ø8 to NA BUS Ø8, ALLOW RETRY inclusive or with NA FIELD Ø9 to NA BUS Ø9, NA FIELD 1Ø to NA BUS 1Ø, FPA RESPONSE inclusive or with NA FIELD 11 to NA BUS 11

24 DISP J > 44: NA FIELD Ø:Ø9 to NA BUS Ø:Ø9, not J MUX > 44 and PC SECT # not = Ø inclusive or with NA FIELD 1Ø to NA BUS 1Ø, NA FIELD 11 to NA BUS 11

25 DISP IBOX TRAPA: NA FIELD Ø:Ø7 to NA BUS Ø:Ø7, I, OP2, OP3 PF inclusive or with NA FIELD Ø8 to NA BUS Ø8, INSN EQUAL AC inclusive or with NA FIELD Ø9 to NA BUS Ø9, INDIRECT inclusive or with NA FIELD 1Ø to NA BUS 1Ø, not XR INVALID inclusive or with NA FIELD 11 to NA BUS 11

26 DISP EACALC 2: NA FIELD Ø:Ø7 to NA BUS Ø:Ø7, EA BUFFER Ø6:17 = Ø inclusive or with NA FIELD Ø8 to NA BUS Ø8, not Y REG bit Ø inclusive or with NA FIELD Ø9 to NA BUS Ø9, not Y REG bit Ø1 inclusive or with NA FIELD 1Ø to NA BUS 1Ø, Y REG bits 2:5 not = Ø inclusive or with NA FIELD 11 to NA BUS 11

27 DISP EACALC 1: NA FIELD Ø:Ø7 to NA BUS Ø:Ø7, EA BUFFER Ø6:17 = Ø inclusive or with NA FIELD Ø8 to NA BUS Ø8, Y REG Ø0,Ø6:17 < or = Ø inclusive or with NA FIELD Ø9 to NA BUS Ø9, not Y REG bit 13 inclusive or with NA FIELD 1Ø to NA BUS 1Ø, Y REG bits 14:17 not = Ø inclusive or with NA FIELD 11 to NA BUS 11

30	DISP EARLY DCD 0 and 2: NA FIELD 0:7 to NA BUS 0:7, ONES to NA BUS 8:9, EARLY DECODE 00 (OP2 needed) inclusive or with NA FIELD 10 to NA BUS 10, EARLY DECODE 02 (write test) inclusive or with NA FIELD 11 to NA BUS 11
31	PXCT CHECK: NA FIELD 0:8 to NA BUS 0:8, not PREVIOUS ENABLE inclusive or with NA FIELD 09 to NA BUS 09, not Y REG 0 inclusive or with NA FIELD 10 to NA BUS 10, ALU 0:35 not equal zero inclusive or with NA FIELD 11 to NA BUS 11
32	SPARE 2 WAY DISP (IMPLEMENTED) inclusive or with NA FIELD 10 to NA BUS 10, USE AC 0 inclusive or with NA FIELD 11 to NA BUS 11
33	DISP PXCT ENABLED: NA FIELD 0:7 to NA BUS 0:7, PXCT ENABLED inclusive or with NA FIELD 08 to NA BUS 08, NA FIELD 9:11 to NA BUS 9:11
34	DISP NOT SC REG 2: NA FIELD 0:10 to NA BUS 0:10, not SC REG 02 inclusive or with NA FIELD 11 to NA BUS 11
35-36	spare
37	DISP INTERRUPT: NA FIELD 0:7 to NA BUS 0:7, ONES to NA BUS 08:09, 1MSEC REQ inclusive or with NA FIELD 10 to NA BUS 10, REQ INT TRP inclusive or with NA FIELD 11 to NA BUS 11
40-57	RETURN: Return stack 0:10 inclusive or with NA FIELD 0:10 to NA BUS 0:10, skip bit determined by NA SWITCH decodes 00:17 inclusive or with Return stack 11 or NA FIELD 11 to NA BUS 11

60 NA FIELD Ø:7 to NA BUS Ø:7, Instr AC addr inclusive or with NA FIELD 8:11 to NA BUS 8:11

61 NORMALIZE: NA FIELD Ø:7 to NA BUS Ø:7, ALU Ø:35 equal Ø inclusive or with NA FIELD Ø8 to NA BUS Ø8, F BUS 8,9, not F BUS bit -1 inclusive or with NA FIELD 9:11 to NA BUS 9:11, respectively. (same timing as decode 7)

62 ENORMALIZE: NA FIELD Ø:7 to NA BUS Ø:7, ALU Ø:35 equal Ø inclusive or with NA FIELD 8 to NA BUS 8, F BUS 11, 12, not F BUS bit -1 inclusive or with NA FIELD 9:11 to NA BUS 9:11, respectively. (same timing as decode 7)

63 DIVIDE: NA FIELD Ø:8 to NA BUS Ø:8, not SC REG bit Ø, not X REG bit Ø, CARRY OUT of ALU inclusive or with NA FIELD 9:11 to NA BUS 9:11, respectively. (same timing as decode 7)

64 PI ENC: NA FIELD Ø:8 TO NA BUS Ø:8, INTERRUPT ENCODE inclusive or with NA FIELD 9:11 to NA BUS 9:11

65 MUL1: NA FIELD Ø:8 to NA BUS Ø:8, not SC REG Ø, MQ REG 34 and 35 inclusive or with NA FIELD 9:11 to NA BUS 9:11

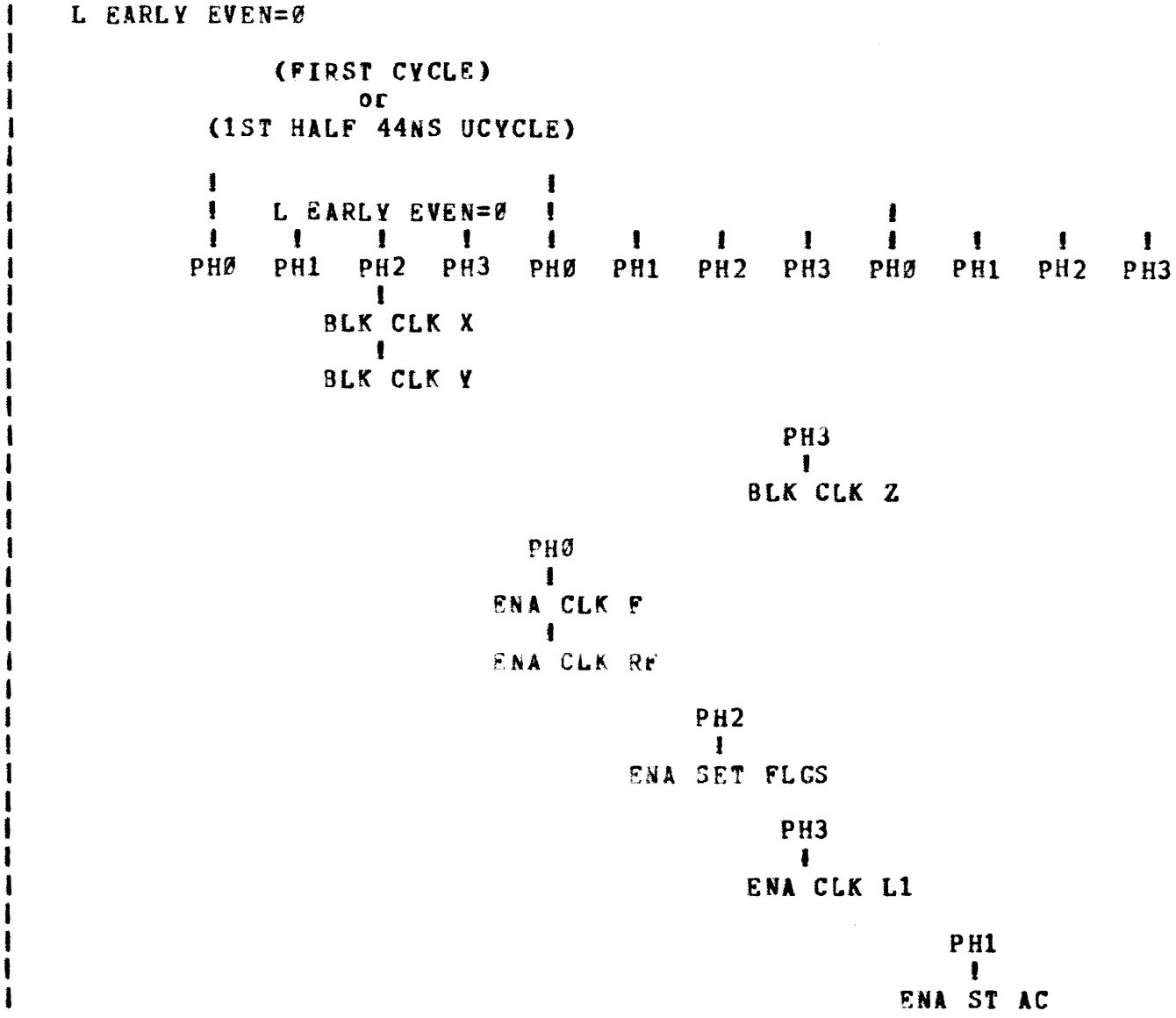
66 BYTE: NA FIELD Ø:8 to NA BUS Ø:8, F BUS 12 anded with PC sect not=Ø, not SCAD ALU bit Ø, FIRST PART DONE flag inclusive or with NA FIELD 9:11 to NA BUS 9:11

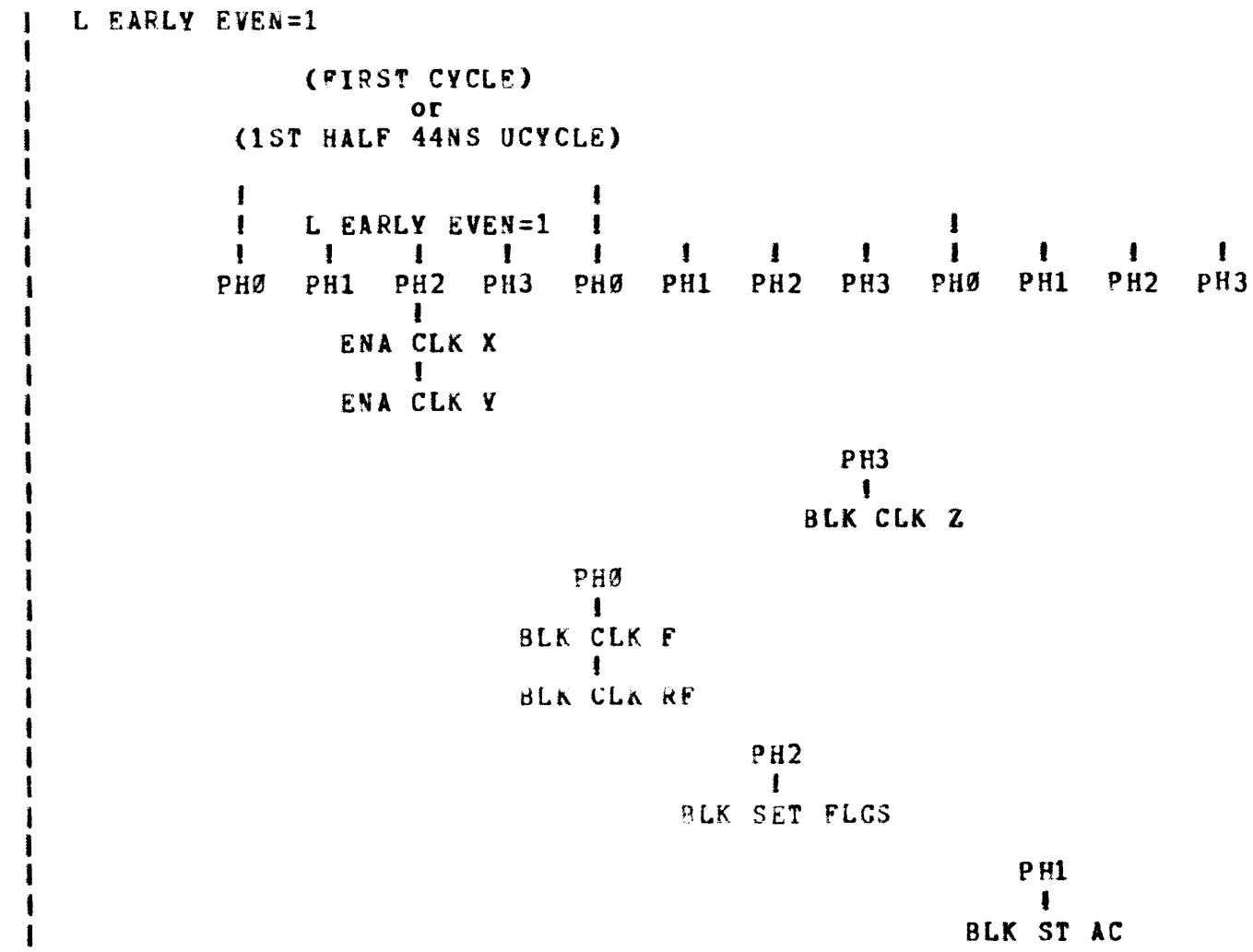
67 EARLY DECODE: NA FIELD ØØ:Ø7 to NA BUS ØØ:Ø7, OP3 is GLOBAL inclusive or with NA FIELD Ø8 to NA BUS Ø8, EARLY DECODE 4:6 inclusive or with NA FIELD Ø9:11 to NA BUS Ø9:11

70	SIGNS: NA FIELD Ø:9 to NA BUS Ø:9, not X REG Ø, not Y REG Ø inclusive or with NA FIELD 10:11 to NA BUS 10:11
71	RETRY: NA FIELD Ø:8 to NA BUS Ø:8, PIPE ERROR (FRU REG Ø1 OR Ø8) inclusive or with NA FIELD 8 to NA BUS 8, RETRY CODE <FRU REG 12:13> inclusive or with NA FIELD 10:11 to NA BUS 10:11
72	MUL2: NA FIELD Ø:8 to NA BUS Ø:8, not SC REG Ø, MQ REG 32, MQ REG 33 inclusive or with NA FIELD 9:11 to NA BUS 9:11
73	FLAGS MUX: NA FIELD Ø:8 to NA BUS Ø:8, not FLAGS MUX 1:3 inclusive or with NA FIELD 9:11 to NA BUS 9:11
74	IC LO: NA FIELD Ø:7 to NA BUS Ø:7, INSN CODE bits 5:8 inclusive or with NA FIELD 8:11 to NA BUS 8:11
75	Y REG: NA FIELD Ø:7 to NA BUS Ø:7, not Y REG bits Ø:3 inclusive or with NA FIELD 9:11 to NA BUS 9:11
76	EM MODE: NA FIELD Ø:7 to NA BUS Ø:7, ONE to NA BUS 8, EM MODE inclusive or with NA FIELD 9 to NA BUS 9, ONES to NA BUS 10:11
77	IC HI: NA FIELD Ø:7 to NA BUS Ø:7, INSN CODE bits 1:4 inclusive or with NA FIELD 8:11 to NA BUS 8:11

2.6.21 L EARLY EVEN (UW Bit 64)

The L EARLY EVEN bit is selected into a control line known as L EARLY during First CYcle and also during the first half of any 44ns microcycle. L EARLY EVEN gates data path clocking as follows:


L EARLY EVEN=Ø
(FIRST CYCLE)
or
(1ST HALF 44NS UCYCLE)
L EARLY EVEN=Ø
PHØ PH1 PH2 PH3 PHØ PH1 PH2 PH3 PHØ PH1 PH2 PH3
BLK CLK X
BLK CLK Y
PH3
BLK CLK Z
PHØ
ENA CLK F
ENA CLK RF
PH2
ENA SET FLGS
PH3
ENA CLK L1
PH1
ENA ST AC



2.6.22 SC FIELD (UW Bits 65:67)

The SC field controls the SC REG and SC MUX on the SCA module. UW bit 65 on causes a hold of previously clocked contents of the SCAD ALU output. UW bit 65 off allows SC REG to clock. UW bits 66:67 decode as follows:

decode	function
Ø	SC REG 6:11 to SC MUX Ø:5
1	SC ALU 6:11 to SC MUX Ø:5
2	# FIELD 12:17 TO SC MUX Ø:5
3	FPA SHIFT Ø:5 to SC MUX Ø:5

If the selected source of the SC MUX is either the SC REG or the SC ALU, and bits Ø:11 of the selected source are greater than the value 44(octal), the SC MUX output is forced to 44. The # field and FPA shift values will never exceed 44.

2.6.23 FE FIELD (UW Bits 68:69)

The FE field controls the FE REG and FE MUX on the SCA module. UW bit 68 on causes a hold of previously clocked contents of the FE MUX output. UW bit 68 off allows FE REG to clock. UW bit 69 selects FE MUX as follows:

UW Bit 69 off K MUX Ø:11 to FE MUX Ø:11
UW Bit 69 on SCAD ALU Ø:11 to FE MUX Ø:11

2.6.24 CALL (UW Bit 7Ø)

The CALL microorder will cause the address dispatch stack to be loaded with the value of the previous NA BUS. The AD STACK is a 16 word LIFO register file, and may be read by the RETURN microorder.

2.6.25 J FIELD (UW Bits 71:73)

The J field controls the J input to the SC ALU <SCA module> via the J MUX. The field decodes are as follows:

decode	function
Ø	SC REG Ø:11 to J MUX Ø:11
1	# FIELD Ø:11 to J MUX Ø:11
2	X REG 18 TO J MUX Ø:3, X REG 28:35 to J MUX 4:11
3	FE REG Ø:11 to J MUX Ø:11
4	zeros to J MUX Ø:5, Y REG Ø:5 to J MUX 6:11
5	zeros to J MUX Ø:7, Y REG 14:17 to J MUX 8:11
6	zeros to J MUX Ø:7, instruction XR value 14:17 to J MUX 8:11
7	zeros to J MUX Ø:7, Y REG 2:5 to J MUX 8:11

2.6.26 K FIELD (UW Bits 74:76)

The K field controls the K input to the SC ALU <SCA module> via the K MUX. The field decodes are as follows:

decode	function
Ø	# FIELD Ø:11 to K MUX Ø:11
1	zeros to K MUX Ø:5, Y REG 6:11 to K MUX 6:11
2	zeros to K MUX Ø:5, FPA SHF ENCODE to K MUX 6:11
3	zero to K MUX Ø, OE Y REG Ø with Y REG 1:8 to K MUX 1:8, zeros to K MUX 9:11
4	zero to K MUX Ø, OE Y REG Ø with Y REG 1:11 to K MUX 1:11
5	FE REG Ø:11 to K MUX Ø:11
6	zeros to K MUX Ø:11
7	ones to K MUX Ø:11

2.6.27 SC ALU (UW Bit 77)

The SC ALU bit determines the SC ALU function:
Ø=add K MUX Ø:11 to J MUX Ø:11
1=subtract K from J

2.6.28 T (UW Bit 78)

T bit on will causes the current microword to hold for 22ns, allowing the next microword's address additional setup time (late skips or dispatches are handled in this way).

2.6.29 PARITY CRA (UW Bit 79)

PARITY CRA is turned on or off so that odd parity is achieved for the 51 microcode bits located on the CRA module (27:28,31:32,34:36, 41:79,1ØØ:1Ø4).

2.6.30 NUMBER FIELD 0:17 (UW Bits 80:97)

The number field is an 18 bit field which is used in conjunction with other microorders as a direct input into data or control logic.

2.6.31 # FIELD PARITY BITS Ø:8 (UW Bit 98)

The # FIELD PARØ is turned on or off so that odd parity is achieved for number field bits Ø:8 (UW bits 8Ø:88). UW bit 98 is included in the overall ESE parity check.

2.6.32 # FIELD PARITY 9:17 (UW BIT 99)

THE # FIELD PAR1 is turned on or off so that odd parity is achieved for number field bits 9:17 (UW bits 89:97). UW bit 99 is included in the overall ESE parity check.

2.6.33 D FIELD (UW Bits 1ØØ:1Ø2)

The D field controls the D MUX at the SHL and SHR modules. If L EARLY is on, the D FIELD controls MQ operation during the 2nd half of the current microcycle. If L EARLY is off, the D FIELD controls MQ operation during the first half of the current microcycle. If L EARLY is on and a write to X or Y REG files is given, the current microcycle's D FIELD will determine the data to be written. If L

EARLY is off and a write to X or Y REG files is given, the next microcycle's D FIELD will determine the data to be written. The field decodes are as follows:

decode	function
Ø	L1 REG Ø:35 to D MUX Ø:35
1	L1 REG 1:35 to D MUX Ø:34, MQ REG Ø to D MUX 35,MQ REG 1:35 to MQ REG Ø:34, ALU overflow to MQ REG 35
2	L1 REG Ø:34 to D MUX 1:35, L1 REG 35 to MQ REG Ø, F BUS -Ø1 to D MUX Ø, MQ REG Ø:34 to MQ REG 1:35
3	L1 REG Ø:33 to D MUX 2:35, L1 REG 34:35 to MQ REG Ø:1, F BUS -Ø2 t D MUX Ø, F BUS -Ø1 to D MUX 1, MQ REG Ø:33 to MQ REG 2:35
4	Z MUX Ø:35 to D MUX Ø:35
5	Z MUX 6:35 to D MUX 6:35, SCAD ALU 6:11 to D MUX Ø:5
6	Z MUX 9:35 to D MUX 9:35, SCAD ALU 4:11 to D MUX 1:8, zero to D MUX Ø
7	Z MUX 12:35 to D MUX 12:35, SCAD ALU 1:11 to D MUX 1:11, zero to D MUX Ø

2.6.34 MQ (UW Bit 103)

The MQ bit determines the mode of the MQ REG. If off, the MQ REG performs a parallel load with data from the L1 REG. If on, the MQ REG shifts per D MUX control (if the D MUX performs a non shift operation, MQ holds it's value).

2.6.35 RF (UW Bit 104)

The RF bit determines selection of either R BUS or A BUS to the L MUX input at the MVL and MVR modules. First cycle control of this field forces selection of R BUS. Since the RF bit is a late control, the results of the move path operation during first cycle are selected into the RF BUS. The decode is as follows:

Ø= R BUS Ø:35 to RF BUS Ø:35
1= A BUS Ø:35 to RF BUS Ø:35

2.6.36 L EARLY ODD (UW Bit 105)

The L EARLY ODD bit is selected into the control line L EARLY during the the second half of any 44ns microcycle, and gates data path clocking in the same manner as L EARLY EVEN (UW bit 64).

2.7 EBOX INTERFACE TO IBOX

2.7.1 Inputs To EBOX From IBOX

2.7.1.1 AC ADR FLD Ø9:12 (source INX1, IDLE) -

At the LAST CYCLE of an instruction, a new SELI is generated which will read the AC fields of the two possible successor instructions to that SELI. During the execution cycles of SELI (1st cy through last cy) the AC ADR FLD will contain the AC FIELD of the predicted successor instruction. At LAST CYCLE of SELI, AC ADR FLD Ø9:12 will be clocked into the Y AC RD ADR latches at SHL and SHR. (dest SHLF, SHRF)

2.7.1.2 AC REF TO ESE (source CFL) -

Indicates that the address on the VMA is an AC address as opposed to a memory address. True if an instruction address and VMA 18:31 = Ø, if an IBOX local fetch and VMA 18:31 = Ø, or if VMA Ø6:16 and 18:31 = Ø. At ESE gates the write to AC sets for a write memory (which is AC REF), and enables N+1 forwarding compare at CFL. (dest ESEJ)

2.7.1.3 DIAG CLK PAS 12 (source IDRM) -

Diagnostic clock for the CPU passive scan path.(dest CRAA)

2.7.1.4 E PF (source MID3) -

Page fail indication for EBOX memory reference. (dest CRAG)

2.7.1.5 EA BUF Ø6-17=Ø (source CFL) -

Bits Ø6:17 = zero of the EA BUF addressed by the current EBOX SELI. Used as a branch input to EBOX microcode next address.(dest CRAG)

2.7.1.6 EARLY DCD ØØ, Ø2, Ø4, Ø5, Ø6 (source ISQ) -

Information decoded off the instruction code at instruction fetch time by the IBOX, is addressed by the EBOX SELI during EBOX execution, and used as branch inputs to the EBOX microcode next address. (dest CRAG)

2.7.1.7 EBOX STEP A (source CFL Via CRAI) -

EBOX STEP from the IBOX indicates to the EBOX that the 1st cycle in progress is a valid operation, that all information needed to begin execution of the instruction was prefetched and presented properly to the EBOX by the IBOX. At CRAH it is used to gate the clock of the GLOBAL and PC SECT = Ø indicators. At SCAJ it is used to gate the update of the FLAG HISTORY BUFF address. (dest SCAJ)

2.7.1.8 FORCE LO L CLK (source IDRH) -

This signal is strobed by diagnostic controls at IDR. It allows results from the ALU or MOVE path to be gated through the L MUX while EBOX microcode does not advance. Thus a single micro-word causing a MOVE or ALU operation to occur may be issued, and results may be routed into and scanned at L REG. Normally, L MUX may not be clocked until CLK 1 of the machine cycle following the MOVE or ALU operation. (dest MVLM, MVRM)

2.7.1.9 HOLDOFF TO SHL, SHR (source MIDA) -

With WR 3 UW 36 off, Holdoff to SHR will cause VMA 32:35 to be selected into the 4 hi-order bits of the Master AC address. Holdoff to SHL, SHR will occur due to holdoffs which are the effect of EBOX waiting for a) IBOX to respond to a SPFN ICMD, b) an MBOX access to complete, represented by MBOX RESPONSE or ABORT CYCLE. (dest SHLF, SHRF)

2.7.1.10 OP2 AC ADR 32:35 (source CFL) -

The address of the X AC set supplied by the IBOX.(dest SHRE, SHLE)

2.7.1.11 OP2 BUS (source IDL1-5, IDR1-5) -

Timing: lo-order bit must be valid T2+1.Ø to T2+12.7 /LU hi-order bit must be valid T2+2.Ø to T2+13.7 /LU 36 bit data path from IBOX to EBOX. It may be selected into the X1 MUX and thus into SEL OP2 via the EBOX microcode X1 field control bits. (dest MVE1-5)

2.7.1.12 VMA 32:35 (source IDRC,D) -

Lo-order bits of the virtual memory address; is used to access the ACs in the event that a VMA access of the MBOX represents an AC address.(dest SHRE, SHLE)

2.7.1.13 X1 EN (source IDLE) -

From ISET microcode in the IBOX, is used to force the output of the X1 MUX to zeros during the 1st cycle of selected instructions. See IBOX MICROCODE CONTROL SPECIFICATION, ISET, under X1 DISABLE. (dest MVE1)

2.7.1.14 Y1 EN (source IDLE) -

From ISET microcode in the IBOX, is used to force the output of the Y1 MUX to zeros during the 1st cycle of selected instructions. See IBOX MICROCODE CONTROL SPECIFICATION, ISET, under Y1 DISABLE. (dest MVE1)

2.7.2 Outputs From EBOX To IBOX

2.7.2.1 # FLD 00:17 UW 80:97 (source ESEE) -

Input to J MUX, K MUX, flags on SCA. # FLD bits 00:13 go to MID, latched by SPFN ICMD, as qualifiers for memory operations. # FLD bits 14:17 go to ISQ as the 4 lo-order bits of the IPUT microcode address of the 1st IPUT microword of an EBOX routine. See EBOX MICROCODE CONTROL SPECIFICATION under '# FLD' and 'SPFN ICMD'. (dest SCA2, MID8, ISQH)

2.7.2.2 ALLOW RETRY (source SCAH) -

SPFN REG bit which, if off, indicates to the console, following a hardware error, that the instruction in execution had entered a region of microcode which precluded instruction retry. Allow retry is set on at Clock 3 of Last Cycle. (dest CRAG, INXC)

2.7.2.3 BLOCK STEP (source SCAH) -

Signal to the IBOX which will prevent the loading of a new SELI upon the occurance of the next Last Cycle, causing EBOX to remain in the current SELI. This allows instruction execution microcode at EBOX to be entered at the start of an instruction following the handling of setup due to page fail traps, the 'execute' instruction, etc. (dest CRAF, CFL)

2.7.2.4 CLR JREAL (source SCAH) -

Causes the clear of the Jump Real latch at IDLI. For certain instructions, ISET (IDL) in 1st cycle is not capable of determining the true status of the Jump Real latch. In these cases, ISET will set JUMP REAL on unconditionally. Later, when the EBOX has determined the

true state of the jump, CLR JREAL is used to clear the Jump Real latch if needed. (dest CFL)

2.7.2.5 COMP ØØ, Ø9 (source MVE6) -

COMP ØØ is the "and" of SEL OP1 bit ØØ and SEL OP2 bit ØØ. COMP Ø9 is the "and" of SEL OP1 bit Ø9 and SEL OP2 bit Ø9. The "and" of the two bits is gated to IBOX jump determination logic if the EBOX microcode R/ Y REG RD field selects the COMPARE BUS into the R MUX. (dest IDLI)

2.7.2.6 COMP Ø1-Ø8, 1Ø-17, 18-26, 27-35 EQ (source MVEM) -

The equal result of a compare of the COMP BUS to Ø, -1, or +1. These signals are sent to IBOX jump determination logic. The value being compared against is determined by the EBOX microcode COMP field. (dest IDLI)

2.7.2.7 COMP Ø1-Ø8, 1Ø-17, 18-26, 27-35 G (source MVEM) -

These signals denote that a segment of the COMP BUS is greater than the value Ø, -1, or +1. They are sent to IBOX jump determination logic. The value being compared against is determined by the EBOX microcode COMP field. (dest IDLI) h13 COMP Ø1-Ø8, 1Ø-17, 18-26, 27-35 LS (source MVEM)

These signals denote that a segment of the COMP BUS is less than the value -1 or +1, as determined by the EBOX microcode COMP field. They are sent to IBOX jump determination logic. (dest IDLI)

2.7.2.8 DIAG SH PAS A (source IDRM) -

When high, the passive scan paths of the CPU are in serial shift mode. When low, the passive scan paths are in parallel load mode. (dest CRAA, IDRM, MIDl)

2.7.2.9 EBOX AC Ø9:12 (source CFL) -

The AC field of the instruction currently in execution by the EBOX. (dest SCAH, MIDC, CRAG)

2.7.2.10 EBOX EACALC (source CFL) -

Causes an IBOX trap to the EBOX microcode in the 1st cycle of a new EBOX instruction. Also is used as a branch input to EBOX microcode, indicating that the IBOX was unable to perform the correct effective address calculation. (dest CRAG)

2.7.2.11 EBOX NO JUMP (source SCAB) -

For FLAG FLD decode 10, is an indication that none of the flags CRY0, OV, CRY1, or FOV were cleared. It goes via CFL to IDLI where the JUMP REAL latch is reset. (dest CFL)

2.7.2.12 EBOX XR 14:17 (source CFL) -

The index register field of the instruction currently in execution. It is valid during an EBOX routine at the IBOX. (dest SCA3)

2.7.2.13 FORCE 1ST TO CRA, ESE (source INXG) -

Signal from IBOX that indicates that the IBOX has not completed the prefetch of information necessary for the EBOX to execute the instruction. Causes ESE and CRA to hold in 1st cycle, while blocking destructive stores that may be issuing from the EBOX control store. (dest CRAH, ESEK)

2.7.2.14 FORCE 1ST TO SCA (source CFL Via CRAH) -

Force 1st will prevent the update of flags on SCA, and control the the setting of ODD and EVEN CYCLE latches at ESE and CRA. Force 1st indicates that the instruction is not valid, that the IBOX not yet completed the prefetch of information necessary for the EBOX to execute the current instruction. (dest SCA7)

2.7.2.15 GLOBAL A (source CFL) -

This indicator is set by the initial EACALC that the IBOX performs on an instruction. It may be changed under microcode control between the time of the 1st EACALC and EBOX execution. It is selected into the GLOBAL signal to MVL for operation in carry control logic of the main ALU. See EBOX MICROCODE CONTROL SPECIFICATION under 'CARRY CONTROL'. (dest CRAH)

2.7.2.16 I,OP2,OP3 PF (source MID3) -

A trap to the EBOX microcode at 1st cycle, and branch input to EBOX microcode next address, indicating that a page fail has occurred on an IBOX prefetch of instruction or data for the current SELI. (dest CRAG)

2.7.2.17 IBOX RESPONSE (source ISQ) -

Branch input to EBOX next address, from IBOX microcode special function. (dest CRAG)

2.7.2.18 INCREM TBC (source MIDB) -

1 micro-second interval signal which is synchronized to the CPU clocks by logic on MID. This signal causes the Time Base Counter update by +1. (dest SCAB)

2.7.2.19 INDIRECT (source MID2) -

A trap to EBOX microcode at 1st cycle, and a subsequent branch input to EBOX microcode next address, indicating that the instruction's indirect bit (13) is on. (dest CRAG)

2.7.2.20 INSN CODE 0:8, P0-8 (source IDLE) -

Instruction code of the next instruction to be executed in the EBOX. During last cycle it addresses the ESE fast rams for access of the 1st cycle's microword. Also during last cycle it is latched and held at CRA for access of the 2nd microcycle's control word. (dest ESE1)

2.7.2.21 INT TIM +1 (source MIDB) -

10 micro-second interval signal which is synchronized to the CPU clocks by logic on MID. This signal causes the update of the Interval Timer. (dest SCA8)

2.7.2.22 L BUS (source MVED, E, F) -

The L BUS is the 36 bit result data path of the EBOX. It is used to send data to memory or ACs to be stored, to send a memory address and qualifiers to the IBOX, and to re-circulate data back through the SHL, SHR data paths. It is selected from the F BUS, R BUS, or A BUS, under

EBOX L and RF control bits. (hi dest=SHLJ, SHRJ, FPA, lo dest=IDLA, IDRG, MDPN)

2.7.2.23 MAGIC NUM 14:17 (source SCA2) -

FIELD bits used to increment the OP2 AC Address by required offset. See EBOX MICROCODE CONTROL SPECIFICATION under "WR CTRL". (dest CFL)

2.7.2.24 MVL, MVR ALU ERR (source MVEL) -

Indicates that the propagates and generates of the duplicate ALUs were not equal. Once set, it will hold until CLEAR ERRS is issued under diagnostic control. It is also latched and held at the register inputs to the FRU logic at MID. Providing no other higher priority failure has occurred at the same time, ALU ERR will set an FRU code indicating an MVE failure. (dest MID1)

2.7.2.25 MVL, MVR BACKUP ERR (source MVEK) -

Indicates that bad parity was detected at the output of the MAC backup RAMs during a non-write cycle. (dest CFL)

2.7.2.26 MVL, MVR INPUT ERR (source MVEK) -

Indicates that bad parity was detected at SEL OP1 or SEL OP2. It will cause a freeze of the SEL OP1, SEL OP2 latches and ALU outputs. It is latched and held at the register inputs to the FRU logic at MID. If an X BUS, Y BUS or OP2 BUS, or other higher priority failure has not occurred at the same time, the FRU code will indicate an MVE failure. (dest MID1)

2.7.2.27 MVL, MVR OUTPUT ERR (source MVEK) -

Indicates that bad parity was detected on the L BUS. It will cause a freeze of the L BUS by blocking further clocks to the L MUX. It will also block writes to the MAC backup RAMs. It is sent to MID where it is latched and held at the register inputs to the FRU logic. Providing no other higher priority failure has occurred at the same time, the FRU code will indicate an MVE failure. (dest MID1)

2.7.2.28 PCU/USER IO FLAG (source SCAG) -

PCU/USER IO FLAG goes to CRA for next address branching: to MID for control of previous context memory access and determination of VMA Ø5. (dest CRAI, MIDC)

2.7.2.29 USER FLAG (source SCAG) -

USER FLAG to EBOX next address dispatch at CRA, and to previous context memory access selection and VMA Ø5 at MID. (dest CRAI, MIDC)

2.8 EBOX INTERFACE TO THE IOP

2.8.1 Inputs To EBOX From IOP

2.8.1.1 CSL ATTN (source IOP) -

Interrupt from the console which enters the CPU as APR FLAG 2. (dest SCAI)

2.8.1.2 IL 1:5 TO CPU (source IOP) -

Interrupt levels 1 through 5 from the IO PORTS. (dest SCAA)

2.8.1.3 PWR FAIL (source IOP) -

Indication of an AC power failure presented as an interrupt via APR FLAG 3. (dest SCAI)

2.8.2 Outputs To IOP From EBOX

2.8.2.1 ECL BROADCAST (source SCAH) -

IO Register bit 31 loaded by SPFN LDIO, to IO PORTS. See EBOX MICROCODE CONTROL SPECIFICATION under "SP FUNCTION, LD IO" (dest IOP)

2.8.2.2 ECL CTRL Ø:1 ENC (source SCAH) -

IO Register bits 28:29 loaded by SPFN LDIO, to IO PORTS. See EBOX MICROCODE CONTROL SPECIFICATION under "SP FUNCTION, LD IO". (dest IOP)

2.8.2.3 ECL S SEL Ø:3 (source SCAH) -

IO Register bits 32:35 loaded by SPFN LDIO, to IO PORTS. See EBOX MICROCODE CONTROL SPECIFICATION under "SP FUNCTION, LD IO". (dest IOP)

2.8.2.4 RESET FROM CPU (source SCAH) -

IO REG bit 3Ø loaded by SPFN LDPI to IO PORTS. See EBOX MICROCODE CONTROL SPECIFICATION under "SP FUNCTION, LDPI". (dest IOP)

2.9 EBOX INTERFACE TO MBOX

2.9.1 Outputs From EBOX To MBOX

2.9.1.1 EN ADR BREAK (source SCAH) -

Signal from EBOX to MBOX to enable the address break feature during memory references. (dest MBOX)

2.9.1.2 FORCE PHY (source SCAH) -

Signal indicating that concurrent EBOX references to memory are represented by a physical address on the VMA. (dest CAMI)

2.9.1.3 L BUS (source MVED, E, F) -

The L BUS is the 36 bit result data path of the EBOX. It is used to send data to memory or ACs to be stored, to send a memory address and qualifiers to the IBOX, and to re-circulate data back through the SHL, SHR data paths. It is selected from the F BUS, R BUS, or A BUS, under EBOX L and RF control bits. (hi dest=SHLJ, SHRJ, FPA, lo dest=IDLA, IDRG, MDPN)

2.1Ø EBOX INTERFACE TO FPA

2.1Ø.1 Inputs To EBOX From FPA

2.10.1.1 A BUS 00:35, 4P (source FPA) -

36 bit result word from FPA. (dest MVEG)

2.10.1.2 FPA RESPONSE (source FPA) -

Input to EBOX microcode next address branch from FPA.(dest CRAG)

2.10.1.3 HOFF UW CLOCKS (source FPA) -

A signal from the FPA which causes the EBOX microcode to hold in a given microcycle, for use in synchronizing the FPA result on the A BUS to the EBOX acceptance of it. (dest ESEI)

2.10.2 Outputs From EBOX To FPA

2.10.2.1 L BUS (source MVED, E, F) -

The L BUS is the 36 bit result data path of the EBOX. It is used to send data to memory or ACs to be stored, to send a memory address and qualifiers to the IBOX, and to re-circulate data back through the SHL, SHR data paths. It is selected from the F BUS, E BUS, or A BUS, under EBOX L and RF control bits. (hi dest=SHLJ, SHRJ, FPA, lo dest=IDLA, IDRG, MDPN)

CHAPTER 3

IBOX

3.1 GENERAL

The JUPITER IBOX is contained on 4 modules: IDR, IDL, INX and TAG. A module adjacent to IDR, called MID, is utilized as an IBOX-MBOX interface, and FRU callout. The primary purpose of the IBOX is to prefetch from the MBOX operands and other information necessary for the execution of instructions by the EBOX, thus eliminating much of the delay penalty usually associated with cache and memory accesses. It serves as the interface between the EBOX and the MBOX for all memory reads and writes. Although the JUPITER IBOX prefetches instructions in a pipeline fashion, it is a microcode controlled unit, and as such provides much flexibility in arriving at algorithms which provide maximum throughput with minimum hardware cost. The TAG system unique to the JUPITER allows multiple, simultaneous access of the MBOX by the IBOX without the part count or conflict overhead usually associated with a hard-wired pipeline structure.

3.2 IDL,IDR MODULES (CPU BACKPANEL SLOTS 11,12)

3.2.1 IDL,IDR Functionality

3.2.1.1 Memory Address Path - The memory address path is located on the IDL and IDR modules. It's output is the VMA (virtual memory address) bus, a 30 bit bus which is used for any of four types of accesses: a) program counter to read memory, b) instruction stream prefetch to read memory, c) effective address calculation to read or write memory, and d) direct address, address+1, or address+2 from EBOX or INDEX AC to read or write memory.

3.2.1.2 Index AC - A copy of the current AC set is maintained on IDL and IDR, and is updated from the L BUS each time the EBOX stores into its current AC set or loads a new set from the MASTER AC block. This AC set is used primarily for EACALCs, where the index reg is added to the Y portion of the instruction (bits 18:35), according to the algorithm described on page 1-22 of the hardware reference manual. It is also used for the memory address determination of the stack instructions PUSH, POP, and PUSHJ.

3.2.1.3 EA ALU - The EA ALU produces a 30 bit binary add, add+1, or add+2 in 22ns. During instruction stream prefetching, the EA ALU is used as the incrementer of the stream addresses, with the value 1 forced into the E2 MUX and the output of the stream buffer (via VMA) routed in to the E1 MUX. In this mode, with a forced carry in, the EA ALU adds 2 to the instruction address to arrive at the next doubleword instruction address boundary. Instructions are fetched 2 at a time from the M BOX from a single VMA address. The results of the stream update are stored in the I STR buffer in addition to being sent to the M BOX over the VMA bus. I stream updates by the EA ALU affect only bits 18:35 of the VMA. Bits 6:17 come from the PC.

When an instruction arrives at the IBOX, a request is made to the IPUT microcode to initiate an EACALC. Provided that the IBOX is not busy with another priority request (such as a command from EBOX), it will initiate an EACALC as a function of the INDEX AC value and the Y portion of the instruction. The results of the EACALC are stored in the EA Buffer, and then sent to the VMA bus for a memory read, to obtain an operand, or to test for page fail of an address representing a write destination. If an MBOX read is not required due to the VMA being an AC address, ABORT CYCLE is sent to the MBOX in the 22ns period following the EACALC. ABORT CYCLE cancels the last IBOX request for an MBOX read. If an MBOX read is not required in the first cycle of the EACALC routine due to OP2 not being needed (EARLY DECODE bit 00), the request is blocked.

The EA ALU is also used to pass an address directly (or +1 or +2) from the INDEX AC or from the EBOX L BUS to the VMA bus. Following a read PC command from the EBOX the EA ALU is used as a path for the EBOX to retrieve the value of the PC (via the VMA bus).

3.2.1.4 I Stream Prefetch - When the PC is loaded after a preset of the IBOX, all 8 buffer slots are empty. The PC will be gated onto the VMA bus to fetch the first instruction. This starting PC value will be tested for a lo-order bit. If on, the EA ALU will increment the VMA +1 to get the next doubleword in the instruction stream. If off, the EA ALU will increment the VMA +2 to get the next doubleword. Each instruction address fetched is stored in the I STR Buffer under a stream address A, B, or C. If no jumps are detected in the instructions coming back, a single stream will be updated, and its most recently addressed location will reside in it's stream buffer word. If a jump out of the stream is detected, a second stream is created, which starts from the the jump target address, is updated and loaded into that stream's word in the STR Buffer.

IPUT (instruction prefetching unit) microcode controls the prefetching of up to 3 separate instruction streams.

3.2.1.5 Conflict Compare - The addresses of both instructions and operands are subject to possible writes by the EBOX. Therefore, indexes are kept to determine whether the 9 lo-order bits of the VMA of a write compare with any instruction addresses or operand addresses of instructions or operands in the I buffers. If a compare occurs against the index, a request is made to IPUT to refetch the contents of the conflicting address. Another comparator checks the AC stores against an INDEX REG index, containing INDEX REG values of all I Buffer entries. If a conflict occurs here, a request is made to IPUT to perform an EACALC with the new INDEX AC value, and refetch the operand.

IBIX contains the PCs of all instructions assigned to the IBOX by IPUT during stream prefetching. IBIX is written with VMA 18:35 when a request is sent to the MBOX followed by a TAG whose source (bits4:5) indicates that the VMA represents an address of a pair of instructions being prefetched.

IBIX performs not only a lo-order 9 bit compare against the instruction address of memory writes, but also an 18 bit lo-order compare against the instruction address of EACALCs resulting from jump instructions. This compare is used by IPUT in determining whether or not a new stream should be created (if a jump target address resides in the I Buffers, no new stream is created).

3.2.1.6 Memory Operand Path - The IBOX prefetches/precalculates the operands associated with all instructions assigned to the IBOX by IPUT. In the case of immediate operands, the EA Buffer will be loaded with the effective address calculation,and a selection is made in the first cycle of EBOX execution at the OP2 MUX of the immediate value, whether it be directly from the EA MUX, or from the EA Buffer. In the case of memory operands, IPUT will, once EACALC has been performed, send the EACALC to the MBOX via the VMA BUS, along with a tag identifying the OP Buffer slot into which returning data will be

written. If the EBOX is in the wait loop (waiting for operand valid bits) when the operand arrives at the IBOX from the MBOX, OP2 MUX selects the MD MUX. If the OP2 valid bit is on, the OP2 MUX selects the OP2 Buffer.

3.2.1.7 OP2 Buffer - The OP2 Buffer is a 16 word register file which is represents the contents of the locations addressed by EACALCs of up to 16 instructions. The OP2 Buffer is read 22ns prior to the start of execution by the EBOX and is gated on to the OP2 bus if the instruction does not require an immediate operand (OP2 is the 2nd operand of an instruction, whereas OP1, the word addressed by the AC field , is the 1st operand of an instruction).

The OP2 Buffer may be loaded from the MD BUS under control of a returning tag whose source bits 4:5 identify an operand fetch or refetch. It is read by SEL I , a 4 bit value which identifies the buffer position of the next instruction to be executed.

3.2.1.8 EA Buffer - The EA Buffer is a 16 word register file which is loaded with the EACALCs (or in the case of STACK instructions the address AC or AC+1) of up to 16 instructions. It is written under control of a tag out of the TAG Buffer. The TAG Buffer is written with the lo-order 4 bits of a re- turning tag if the 3 hi-order bits of the tag identify an instruction fetch or refetch (the TAG Buffer is written by a counter which is then incremented+1, and read by another counter which also incremented +1; counters not equal cause a request to IPUT microcode to initiate an EACALC).

The EA Buffer is read by SEL I, 22ns prior to actual execution of the instruction in the EBOX. It's output will be gated to the OP2 BUS if the instruction requires an immediate operand.

3.2.2 Error And Diagnostic Control

3.2.2.1 Address Path Error Control - Parity is checked at the inputs of the EA ALU. Parity is generated at the output of the EA MUX . One bit parity per 9 data bits is carried for bytes 1, 2, and 3 of the memory address path; one parity bit is carried for bits 6:8 (address byte Ø). The memory address path within the IBOX consists of IBIX, STR Buffer, EA Buffer, EA MUX, VMA, and E1 MUX. Duplicate copies of the EA ALU are compared against one another as an integrity check of the EA ALU operation.

3.2.2.2 Address Path Diagnostic Control - A passive scan reg for the PC is kept on IDL and IDR. In order to obtain the current PC (PC of the next instruction going to the EBOX), the console must interrupt the IPUT microcode and force it into a routine which will issue a load PC scan micro-order. The PC scan register will then be loaded from the PC section reg (IDL) and IBIX (IDR).

3.2.2.3 Data Path Error Control - The OP2 BUS is parity checked as it leaves IDL and IDR. If bad parity is detected, the contents of the OP2 bus are frozen in the OP2 SCAN REG and one of the two OP2 ERR latches are set (IDL or IDR) at CLK Ø. The IBOX data path carries a parity bit for each 9 bits of data, and consists of the L3 BUS, L3 REG, E1 MUX, MD MUX, OP2 Buffer, OP2 MUX, and OP2 ACTIVE SCAN REG.

A flag received from MBOX is stored in the OP Buffer signifying a page failure. This flag, when the OP Buffer is read to be gated onto the OP2 BUS will cause a page fail trap to occur if the EBOX last cycle bit is on. During long instructions where memory is read in cycles other than first cycle (last cycle delayed 22ns), the EBOX microcode will branch on a dispatch of a TAG returning whose source code indicates an EBOX operation, and a separate signal from MBOX indicating page fail.

3.2.2.4 Data Path Diagnostic Control - The OP2 BUS has an active scan register which is clocked every cycle, and is held if an error is detected on the bus. The L3 REG, which may be selected into the MD MUX, the INDEX AC, or the E2 MUX, is also in the active scan path.