



MATRA
DESIGN SEMICONDUCTOR

CHRIS™ EMBEDDED PROCESSOR UNIT (EPU™)

advance information

February 1988

FEATURES

- 32-Bit RISC Architecture
- 20 MIPS Throughput
- Executes 1 Instruction/Cycle
- 3Kb WCS Instruction Cache
- 96-Bit Instruction Word
- 32-Barrel Rotator (Byte)
- Bit Reversal
- Flexible Host and DMA Interfaces
- Supports Single/Double Precision FPU
- 2901 ALU Superset
- 32-Bit ALU or 2 16-Bit ALUs
- 2910 Sequencer Superset
- 2Kb Data Cache
- 3-Address Generators
- 24-Bit Address Space
- Conditional Register Loading
- Single-Cycle Memory Bus
- 0.8 Micron CMOS Process

DESCRIPTION

The CHRIS EPU is a powerful embedded processor unit (EPU) designed to maximize performance and eliminate bottlenecks in compute-intensive applications such as graphics, DSP, speech processing, etc. CHRIS (Controller Having Reduced Instruction Set) can best be described as an advanced dual processor capable of executing integer operations within its integer execution channel while simultaneously managing floating point operations (e.g. multiply, divide, square root, etc.) in its companion floating point unit (FPU). Unlike conventional processors, the driving force behind CHRIS is a sophisticated RISC engine architecture which can be configured by a host processor for specialized embedded tasks. Alternatively, CHRIS can act as a stand-alone microprocessor. The RISC engine utilizes an enhanced 2901/2910 architecture and executes a full superset of the 29xx instruction set. On-chip data and instruction caches allow for maximum throughput and performance. 1-cycle memory bus does not degrade 20 MIPS performance when external data references are necessary. An entire circuit board has effectively been condensed into a single-chip CHRIS. This was possible due to an advanced sub-micron fabrication technology.

CHRIS' companion FPU uses the IEEE Industry Standard Floating Point Format. 34 bits of the CHRIS instruction word are dedicated to managing the floating point path. One can choose between one of four FPU arrangements depending on whether single or double precision is desired. Additionally, one can choose the FPU based on one or three I/O port(s). (See Choosing The Right FPU).

A unique Multiple Chip Module (MCM) package allows the CHRIS EPU and FPU to be packaged together in a single 144-pin PGA package. The CHRIS EPU is also available in a stand-alone 176-pin PGA package.

PIN DEFINITIONS

$\overline{\text{RESET}}$	<u>Reset</u> must remain low for 256 clocks to completely reset the processor. During this period (1) the Host Control Register is set to default (2) Instruction Cache tag bits are invalidated (3) Host and Cache Controllers are set to idle	$\overline{\text{INT}}$	INT - <u>Interrupt</u> is an output signal used by CHRIS to interrupt the host.
$\overline{\text{CS}}$	<u>Chip Select</u> is used by the host to get the attention of the CHRIS EPU. An active low signal on this line is recognized only when both state-machine controllers are in their idle state. Therefore, code or data miss routines are not preemptable by the host until completed. CS is used whenever the host elects to read/write the on-chip memories or the private external memory used by the CHRIS EPU, through the Host Port. The host can additionally write to the Host Control Register and read the Interrupt Register and/or the sequencer address from the Host Port. If CS is active at the completion of Reset , the host can configure CHRIS to behave as a dedicated slave controller before actual execution begins. If $\overline{\text{CS}}$ is not active at this time, CHRIS defaults to stand-alone-master-microprocessor mode and auto-loads its writable control store (WCS).	$\overline{\text{INTA}}$	INTA - Input signal generated by the host to signal <u>Interrupt Acknowledge</u> . This signal is asserted by the host after the Interrupt Vector Register has been read.
		HD31:0	HD31:0 - <u>Host 32-Bit I/O Data Bus</u> .
		$\overline{\text{AS}}, \overline{\text{DS}}$	AS, DS - <u>Address Stable</u> and <u>Data Stable</u> . Same definition as for host, except output signals to external memory.
		$\overline{\text{MACK}}$	<u>Memory Acknowledge</u> . Same definition as for host, except Memory Acknowledge input signal from external memory.
		$\overline{\text{R/W}}$	<u>Read/Write</u> . Same definition as for host, except output signal from CHRIS to external memory.
		D31:0	External memory 32-bit <u>Data Bus</u> .
		A23:0	External memory 24-bit <u>Address Bus</u> .
		UC1:0	<u>User Conditions</u> . Inputs to CHRIS to present external branch conditions to its microsequencer.
$\overline{\text{HAS}}, \overline{\text{HDS}}$	<u>Host Address Stable</u> and <u>Host Data Stable</u> are input signals to CHRIS used by the host to handshake with CHRIS when transferring data to/from the chip's internal or external memory.	SU8:0	In the 32-bit mode, 9 <u>Special User</u> microcode output bits are available to control external peripherals.
$\overline{\text{HACK}}$	<u>Host Acknowledge</u> is an output used to signal the host that data has been latched on a write cycle or placed on the bus on a read cycle.	USR	In the 32- or 16-bit mode, <u>USER</u> provides an additional microcode output bit to control an external peripheral.
$\overline{\text{HR/W}}$	Input signal from host to <u>Read/Write</u> the CHRIS processor's memories.	HA1:0	<u>Host Address Bus</u> is a 2-bit address from the host address bus to select the internal Host I/O registers of CHRIS.
		CLOCK	20 MHz <u>Clock</u> input.

ARCHITECTURE

A detailed block diagram depicting the internal architecture of the CHRIS processor is shown in Figure 1. The key elements of the diagram are the following:

- (1) Integer ALU
- (2) Address Generators
- (3) Floating Point Bus
- (4) Barrel Rotator (Byte)
- (5) Sequencer
- (6) Instruction Cache
- (7) Data Cache
- (8) Host/DMA Interface
- (9) Host and Cache Controllers
- (10) Instruction Register

CHRIS incorporates a Harvard Architecture with separate data and instruction memory. This allows both data and instructions to be accessed simultaneously yielding a 2X improvement in performance. Unlike conventional microprocessors which support a 2-cycle bus, the CHRIS EPU uses a 1-cycle bus to perform all internal and external bus operations resulting in an additional 2X improvement in performance. Wait states occur if the external memory bus cannot cycle in one clock period.

ALU

The CHRIS ALU executes a full superset of the AMD 2901 as shown in Figures 2, 3, 4. Instructions to the ALU are supplied directly out of microcode such that on each new cycle a new ALU operation may begin. All ALU instructions execute in one cycle. Associated with the ALU is a 16 x 32-bit register file. The address lines for the dual-ported register file are supplied out of microcode. Two operands can be supplied simultaneously out of the register file as inputs to the ALU and the resulting ALU output can be written back to the register file to support 1-cycle read-modify-write operations.

To enhance the standard 2901 architecture, additional ALU instructions are provided to support a barrel rotator which can rotate to any byte boundary in 1-cycle, and the ability to perform arithmetic and logical shifts in the ALU Shifter. (See Figures 5,6).

The ALU can operate as either a single 32-bit ALU or by setting the 32/16 bit (See Figure 9, Bit 3) in the Host Control Register to zero, the ALU is split into 2

16-bit ALUs. In 16-bit mode, four addresses are provided out of microcode to provide the the A and B dual-ported addresses for each of the 16 x 16 register files of ALU A and ALU B. In addition to this, the Destination Control lines of each ALU are provided separately out of microcode. This allows separate register addresses and destination controls to be specified to each of the 16-bit ALUs.

Address Generators

When operating in either 16- or 32-bit mode, a 24-bit address generator is available to linearly index up to 16M words of external memory. The on-chip data cache resides within the first 512 locations of the 16M word memory map. The generated address can be used on-chip to provide an address to the 512 locations of the data cache or is available off-chip by means of a 24-bit external address bus. Thus, data can be moved (a) to/from external memory and the CHRIS EPU, (b) to/from external memory and the FPU, or (c) to/from FPU and CHRIS. (See Figure 7).

In the 16-bit mode, both ALU A and ALU B can become two additional address generators. Three address generators are therefore available in the 16-bit mode. These three address generators are useful when the 3-ported FPU (WTL 33xx) devices are desired. These parts can accept two source operands and provide a destination result every clock cycle. (See Choosing The Right FPU).

Floating Point Bus

CHRIS provides an optimized bus interface to the Weitek WTL 3x32 and 3x64 FPU devices. Up to 40 MFLOPs of throughput using the IEEE Industry Standard Floating Point Format can be achieved. 34-bits of CHRIS' microcode are devoted exclusively to providing the instruction to be executed within the FPU. The FPU code bus is sampled on the rising edge of the clock, allowing a new floating point operation to be initiated on each new clock cycle. An instruction executing within the FPU can be totally independent of operations executing within the CHRIS EPU.

A discussion on FPU choices follows later. Each FPU has its own 32 x 32 register file and divide logic unit.

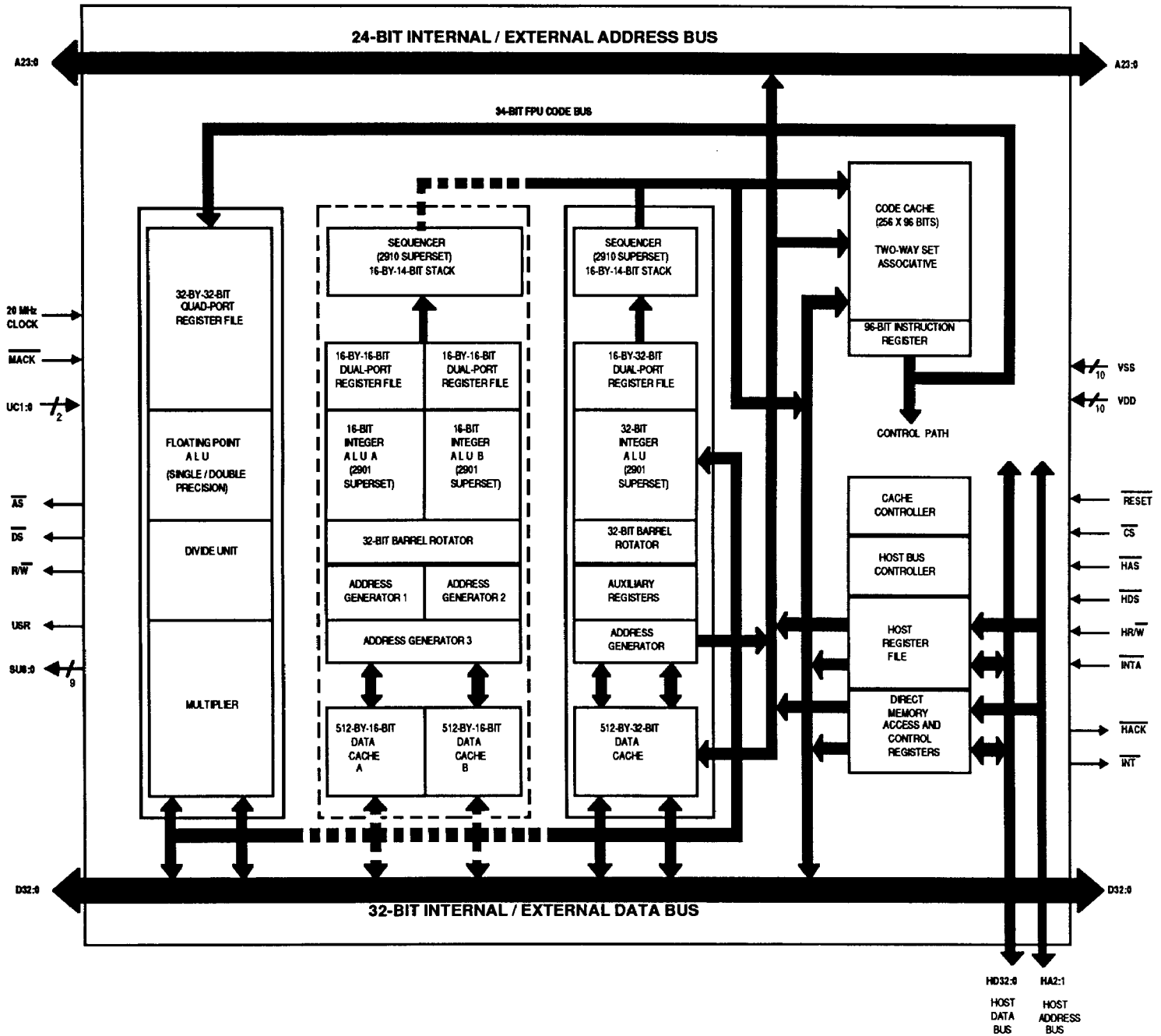


Figure 1. Block Diagram

	Micro Code				ALU Source Operands	
	I2	I1	I0	Octal Code	R	S
AQ	L	L	L	0	A	Q
AB	L	L	H	1	A	B
ZQ	L	H	L	2	0	Q
ZB	L	H	H	3	0	B
ZA	H	L	L	4	0	A
DA	H	L	H	5	D	A
DQ	H	H	L	6	D	Q
DZ	H	H	H	7	D	0

Figure 2. ALU Source Control

Mnemonic	Micro Code				ALU	
	I5	I4	I3	Octal Code	Function	Symbol
ADD	L	L	L	0	R PLUS S	$R + S$
SUBR	L	L	H	1	S MINUS R	$S - R$
SUBS	L	H	L	2	R MINUS S	$R - S$
OR	L	H	H	3	R OR S	$R \vee S$
AND	H	L	L	4	R AND S	$R \wedge S$
NOTRS	H	L	H	5	\bar{R} AND S	$\bar{R} \wedge S$
EXOR	H	H	L	6	R EX-OR S	$R \vee \bar{S}$
EXNOR	H	H	H	7	R EX-NOR S	$\overline{R \vee S}$

Figure 3. ALU Function Control

Mnemonic	Micro Code				RAM Function		Q-REG Function	
	I8	I7	I6	Octal Code	Shift	Load	Shift	Load
QREQ	L	L	L	0	X	NONE	NONE	$F \rightarrow B$
NOP	L	L	H	1	X	NONE	X	NONE
RAMA	L	H	L	2	NONE	$F \rightarrow B$	X	NONE
RAMF	L	H	H	3	NONE	$F \rightarrow B$	X	NONE
RAMQD	H	L	L	4	DOWN	$F/2 \rightarrow B$	DOWN	$Q/2 \rightarrow Q$
RAMD	H	L	H	5	DOWN	$F/2 \rightarrow B$	X	NONE
RAMQU	H	H	L	6	UP	$2F \rightarrow B$	UP	$2Q \rightarrow Q$
RAMU	H	H	H	7	UP	$2F \rightarrow B$	X	NONE

Figure 4. ALU Destination Control

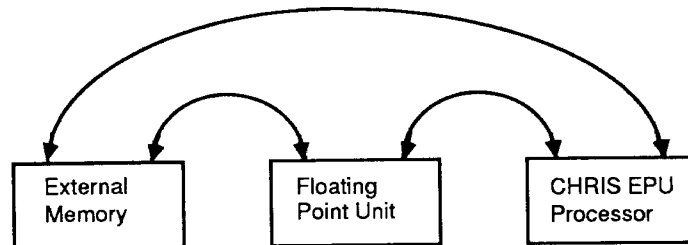


Figure 7. 1-Cycle Data Transfer

Mnemonic	Micro Code		
	Kword	Kbyte	Octal Code
NOP	L	L	0
ROTR 8	L	H	1
ROTL 24	L	H	1
ROTR 16	H	L	2
ROTL 16	H	L	2
ROTR 24	H	H	3
ROTL 8	H	H	3

Figure 5. Barrel Rotator

Mnemonic	Micro Code							ALU FUNCTION
	I8	I7	I6	Octal Code	Kfeed1	Kfeed0	Octal Code	
SHTL0 16	H	H	L	6	L	L	0	Shift Left, LSB ← 0
SHTL1 16	H	H	L		L	H	1	Shift Left, LSB ← 1
SHTLCC 16	H	H	L		H	L	2	Shift Left, LSB ← CC
ROTL 16	H	H	L		H	H	3	Rotate Left
SHTR0 16	H	L	L	4	L	L	0	Shift Right, MSB ← 0
SHTR1 16	H	L	L		L	H	1	Shift Right, MSB ← 1
SHTRCC 16	H	L	L		H	L	2	Shift Right, MSB ← CC
ROTR 16	H	L	L		H	H	3	Rotate Right
SHTL0 32	H	H	L	6	L	L	0	Shift Left, LSB ← 0
SHTL1 32	H	H	L		L	H	1	Shift Left, LSB ← 1
SHTLCC 32	H	H	L		H	L	2	Shift Left, LSB ← CC
ROTL 32	H	H	L		H	H	3	Rotate Left
SHTR0 32	H	L	L	4	L	L	0	Shift Right, MSB ← 0
SHTR1 32	H	L	L		L	H	1	Shift Right, MSB ← 1
SHTRCC 32	H	L	L		H	L	2	Shift Right, MSB ← CC
ROTR 32	H	L	L		H	H	3	Rotate Right

Figure 6. Single Bit Shifter

13-10	MNEMONIC	NAME	REG/CNTR CONTENTS	FAIL CCEN=LOW and CC=HIGH		PASS CCEN=HIGH and CC=LOW		REG/CNTR
				Y	STACK	Y	STACK	
0	JZ	JUMP ZERO	X	0	CLEAR	0	CLEAR	HOLD
1	CJS	COND JSB PL	X	PC	HOLD	D	PUSH	HOLD
2	JMAP	JUMP MAP	X	D	HOLD	D	HOLD	HOLD
3	CJP	COND JUMP PL	X	PC	HOLD	D	HOLD	HOLD
4	PUSH	PUSH/COND LD CNTR	X	PC	PUSH	PC	PUSH	●
5	JSRP	COND JSB R/PL	X	R	PUSH	D	PUSH	HOLD
6	CJV	COND JUMP VECTOR	X	PC	HOLD	D	HOLD	HOLD
7	JRP	COND JUMP R/PL	X	R	HOLD	D	HOLD	HOLD
8	RFCT	REPEAT LOOP, CNTR ≠ 0	≠ 0	F	HOLD	F	HOLD	DEC
			= 0	PC	POP	PC	POP	HOLD
9	RPCT	REPEAT PL, CNTR ≠ 0	≠ 0	D	HOLD	D	HOLD	DEC
			= 0	PC	HOLD	PC	HOLD	HOLD
10	CRTN	COND RTN	X	PC	HOLD	F	POP	HOLD
11	CJPP	COND JUMP PL & POP	X	PC	HOLD	D	POP	HOLD
12	LDCT	LD CNTR & CONTINUE	X	PC	HOLD	PC	HOLD	LOAD
13	LOOP	TEST END OF LOOP	X	F	HOLD	PC	POP	HOLD
14	CONT	CONTINUE	X	PC	HOLD	PC	HOLD	HOLD
15	TWB	THREE-WAY BRANCH	≠ 0	F	HOLD	PC	POP	DEC
			= 0	D	POP	PC	POP	HOLD

● If \overline{CCEN} = Low and \overline{CC} = High, hold; else load.

Figure 8. Sequencer Instructions

Sequencer

The address sequencer for the CHRIS EPU processor supports the full instruction set of the AMD 2910 (See Figure 8). In addition to sequentially addressing the microprogram memory, it provides conditional branching to any microinstruction within its 16K address space. Sixteen levels of nesting are allowed for microsubroutines. A 14-bit register counter provides exit information for microinstruction loops.

The address sequencer provides conditional branching on the true and complement status bits from both integer ALUs and the floating point ALU. Two user-defined external conditions are provided and are selectable through microcode as a branch point to the microsequencer.

Instruction Cache

A fast on-chip instruction cache supports rapid microinstruction fetch. Microinstructions for the CHRIS EPU are stored in its 256 x 96-bit instruction cache (icache). The sequencer presents a 14-bit microaddress to the icache on each clock cycle. The corresponding 96-bit microcoded instruction is loaded into the instruction pipeline register to be presented to the control path at the beginning phase of the next cycle.

The icache is arranged as a two-way-set-associative cache. Each set comprises 32 blocks of 4 microinstructions and a corresponding tag field. The upper 7-bits of the 14-bit microaddress is always written to the tag field whenever data is written to the cache. The eighth bit of the 8-bit tag field is the Valid Bit whose binary value of 1 or 0 determines if the block's microinstructions are valid or not. The stored tag field is addressed and compared to the upper 7-bits of the current microaddress to determine if the instruction lies within the cache. The lower 7 bits of the microinstruction are used to address the cache. Since two sets are involved, two locations with the same base address but differing in tag values can be referenced within the cache.

For many graphics, DSP, or algorithm based application, the on-chip icache provides ample code space for the routine. Applications executing directly in the cache can achieve and maintain the full 20 MIPS performance. For larger applications, a high

speed bus to an external memory, where up to 16M words can be addressed, is provided.

If the address tag for the current instruction is not found within the icache, or if the block Valid Bit is set to invalid, an instruction miss will occur. The Cache Bus Controller will fetch a block of instructions from external memory and perform the necessary cache update transparent to the user program. Special hardware permits single-cycle accesses to external memory provided fast external memory (35nS or less) is used.

CHRIS uses a least-recently-used (LRU) replacement algorithm to decide which half of the icache's set-associative memory to replace on a miss condition. Each block contains its own LRU bit which can be set or reset based on it being least recently used.

During the reset period, each address tag bit is set to invalid. It requires a minimum of 256 clocks to totally invalidate the icache. Invalidation is accomplished by gating a microaddress from the address generator onto the icache. This address generator is incremented until invalidation is completed. When a host executes a reset instruction, it will typically assert reset for a minimum of 256 cycles (512 for the M68xxx). A reset period longer than 256 is fine and simply allows the address generator to roll over and invalidate previously invalidated tags.

Data Cache

Similar to the ALU, the data cache (dcache) can operate as either a 32- or 16-bit cache when instructed by the host. The dcache can act as a scratch pad area or a 512-deep 32-bit register file. Unlike the ALU register file, only one operand per cycle is available in the 32-bit mode. Separate cache addresses in the 16-bit mode allow two operands.

The dcache is positioned in the lower portion of memory from address space 0 to 511. Linear addressing is used throughout the 24-bit address space. Any 9-bit address is guaranteed to "hit" in the dcache. Likewise, any address of 10 bits or greater is guaranteed to result in a "miss" on the dcache.

The addressing on the dcache can be supplied from three sources. In 32-bit mode, the dcache takes the nine LSBs from the 24-bit address generator. In 16-bit mode, each ALU address generator can

provide separate addresses for each half of the dcache. If addresses are further stored in the cache, three separate addresses can be presented to the I/O pads on every cycle. Routines such as Scatter/Gather can be implemented very efficiently. Finally, the host can provide an address to the dcache by way of its Host Address Register to allow direct read/writes.

Host/DMA Interface

A host port interface on CHRIS allows a host computer to communicate with it. The host begins by lowering the CS line to the CHRIS EPU. Two bits of the host address bus are then used to address the host registers within CHRIS.

The host port interface allows the host to read/write the CHRIS' external memory, read/write its data cache, read the sequencer's address for code replacement, and write code directly to the instruction cache. Additionally, the host may configure CHRIS through a

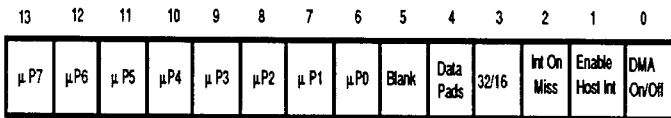


Figure 9. Host Control Register

Host Control Register or read an interrupt vector passed to it by an application program.

The host port supports DMA addressing such that rapid block transfer of data can be made to any of the CHRIS' three memories.

Host and Cache Controllers

Two separate bus controllers provide the handshakes for the external memory bus and the host port. Both controllers are idle in the normal state.

The cache controller (CC) services the miss condition on both caches. On an icache miss, the CC translates the 14-bit sequencer address into a 24-bit memory address to perform the 4-line block replacement. The CC is capable of completing a memory cycle in one clock cycle provided fast static external memory is used. The CC services a miss on the dcache by performing the requested read or write operation from the application program. Again, when fast memory is used, this operation is performed on the same cycle as the miss (1-cycle) so that no degradation of performance is realized. Appropriate

wait states are inserted for slower memories.

A miss condition can be supported even though no external memory is provided. By appropriately setting bit 2 in the Host Control Register, the host indicates that no external memory exists so that the CC will not attempt an external access. Instead, the CC issues an interrupt to the host by asserting INT, thereby allowing the host the ability to provide the necessary instruction code.

An application programmer need not be overly concerned about misses on either cache since servicing them is totally transparent to the executing program.

The host controller (HC) provides the asynchronous handshakes with the host computer and services all host requests. The 25-bit Host Address Register, when written by the host, specifies which of CHRIS' memories will receive the read/write operation including the address within that memory as shown below:

Location	Function
0000000H - 00001FFH	Data Memory
0000200H	Sequencer Address
0000201H - 00003FFH	*** RESERVED ***
0000400H - 00007FFH	Writable Control Store
1000000H - 1FFFFFFFH	External Memory (optional)

Table 1. Internal Memory Map

Bit Reversal

Special on-chip hardware supports address bit reversal for digital signal processing applications. Reversal is performed concurrently with address generation and is available to the address generator on the next clock cycle. Bit reversal can be turned on or off through microcode. See Mode Register in microcode definition section for further details.

Conditional Load

It is often necessary to load a register based on whether a certain condition is true or false. In conventional designs, if the condition is not met, a branch around the register load instruction is made. Up to six parallel paths are supported within the CHRIS EPU. On a given branch, parallel architectures such as CHRIS require consistency in the code to sustain the multiple paths. This may involve duplication of all code necessary to support the parallel channels when branching. The space required to store this additional code can grow proportionately. It is therefore better to execute in-line code than to branch.

The conditional load feature of the CHRIS EPU alleviates the above problems. Conditional load provides for conditional ALU and dcache load operations while maintaining in-line code execution, thereby preserving the parallel paths. Conditional load is turned on and off as needed from within each microcode instruction.

Choosing The Right FPU

Flexible packaging allows for five different arrangements. Table 2 summarizes the possibilities.

Configuration	Companion FPU	I/O Data Ports	Precision	Package	Pins
CHRIS	—	—	—	PGA	176
CHRIS3132	WTL 3132	1	Single	PGA	144
CHRIS 3332	WTL 3332	3	Single	PGA	208
CHRIS3164	WTL 3164	1	Double	PGA	144
CHRIS3364	WTL 3364	3	Double	PGA	208

Table 2. Standard Packages

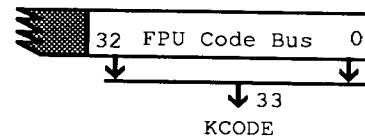
Instruction Register

CHRIS takes advantage of a two-stage pipeline which allows concurrent instruction execution while the next instruction is fetched from the cache. The Instruction Register (IR) contains the instruction which is currently executing and is updated each cycle. The IR

is 96 bits wide and contains over 20 fields. The true performance of the CHRIS EPU is derived from its 20 MIPS throughput times the work performed on each cycle. The extra wide microcode allows many of the features of CHRIS to be addressed in parallel from within each instruction word.

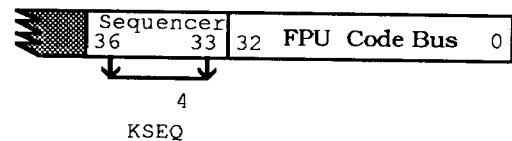
The microcode fields are defined as follows:

Bits 32:0 - FPU Code Field (KCODE)



All actions required to complete a given FPU operation are specified in this 33-bit FPU Code Field. This code bus connects directly to the I/O pads of the CHRIS EPU where new code can be presented to the FPU every cycle. Since the 3-port FPUs require a 34-bit instruction, the user may elect to use user bit 93 (KUSR) to provide the additional bit.

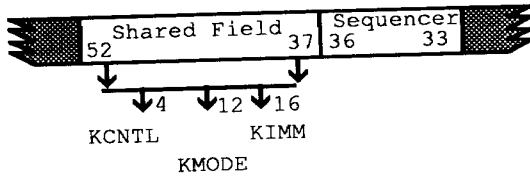
Bits 36:33 - Sequencer Instruction (KSEQ)



The 4-bit instruction (I_3-I_0) for the 2910 sequencer

is provided in this field. See Figure 8 for sequencer instructions.

Bits 52:37 - Shared Field



This 16-bit field is shared between immediate data and set up control since neither are needed every clock cycle. Bit 94 (**KDATA**) makes the selection between immediate data and control.

When Bit 94 is equal to 0, Immediate Data is implied.

Immediate Data (KIMM) - A 16-bit immediate value can be passed directly to the ALU or sequencer. The ALU may use this value to perform arithmetic or logical operations. The sequencer may use this constant value to initialize its internal loop counter or provide the next branch address.

When Bit 94 is equal to 1, Control Mode data is implied.

Mode Register (KMODE) - This 12-bit Mode Register takes its value from bits 48:37. The Mode Register allows internal configuration of the CHRIS EPU through the user program. The Mode Register is loaded with a new value when bits 52:49 are equal to hex 0 as seen below in the Control Mode section. The modes are defined below:

Mode Bits	Action	Function
Mode 0	ALU B Bank A Select ¹	Selects lower eight registers of register file
Mode 1	ALU B Bank B Select	Selects upper eight registers of register file
Mode 3:2	User defined mode bits ²	User defined mode bits
Mode 4	Enable Bit Reverse	Turns address reversal on/off
Mode 11:5	*** RESERVED ***	

¹Applies when ALUs are configured as 2 16-bit ALU.
²These mode bits have been reserved for the end user and are brought out to pins as **UC1:0**.

Table 3. Mode Register Assignments

Control Mode (KCNTL) - Bit 94 equal to 1 implies Control Mode. Bits 52:49 of the shared data field then determine the operation to be performed as seen below:

- 0 NOP
- 1 Control Mode

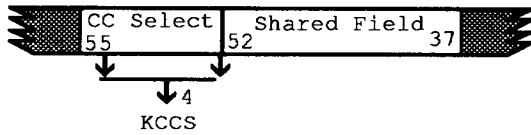
Bits 52:49 Function

- 0000 Load Mode Register (from 48:37)
- 0001 Load Interrupt Vector Register
- 0010 Load Condition Code Register
- 0011 *** RESERVED ***
- 0100 Set Host Interrupt
- 0101 Set Breakpoint
- 0110-0111 *** RESERVED ***
- 1000 *** RESERVED ***
- 1001 Conditional Load ALU A
- 1010 Conditional Load ALU B
- 1011 Conditional Load ALU A & B
- 1100-1111 *** RESERVED ***

Table 4. Mode Register Assignments

Condition Code Polarity (*KCCP*) - Bit 51 controls whether positive or negative condition polarity is to be tested by the sequencer.

Bits 55:52 - Condition Code Select (*KCCS*)

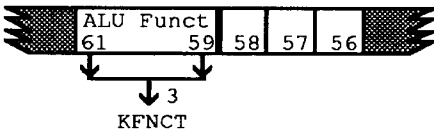


This 4-bit microcode field selects 1 of 16 conditions to present to the sequencer for testing.

Bits 58, 57, 56 - ALU Mux Controls (*KSEL*)

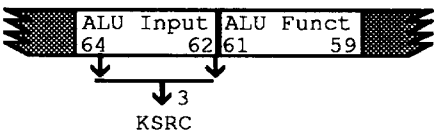
These 3-bits of microcode operate the ALU muxes. *KSELA* and *KSELB* routes either immediate data out of microcode or data off the data bus to the direct inputs of ALU A and ALU B. *KSELC* muxes either immediate data out of microcode or data off the data bus to the direct inputs of the sequencer.

Bits 61:59 - ALU Function Control (*KFNCT*)



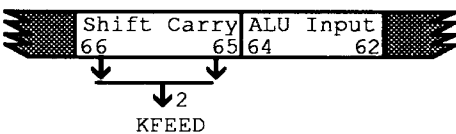
3-bit value which exercises the I_5 , I_4 , and I_3 function controls of the ALU A & B.

Bits 64:62 - ALU Source Control (*KSRC*)



3-bit value which exercises the I_2 , I_1 , and I_0 source controls of the ALU A & B.

Bits 66:65 - Shift/Carry Select (*KFEED*)



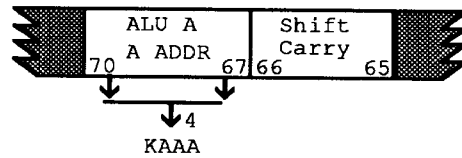
This field is used to determine the input for the ALU shifter and the ALU carry in.

	shift	carry
00	0	0
01	1	1
10	CC*	carry-out (previous cycle)
11	**RESERVED**	**RESERVED**

*Conditions codes can be preserved here.

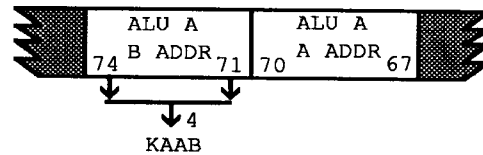
Bits 70:67 - ALU A Address A (*KAAA*)

Provides the A register file address of ALU A.



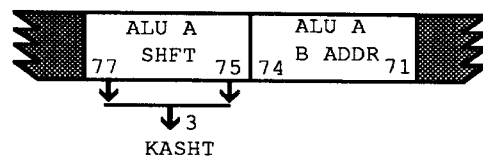
Bits 74:71 - ALU A Address B (*KAAB*)

Provides the B register file address of ALU A.

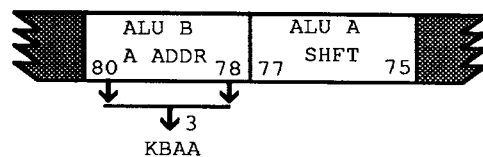


Bits 77:75 - ALU A Shift and Destination Control (*KASHT*)

Provides the shift/destination instruction for ALU A.

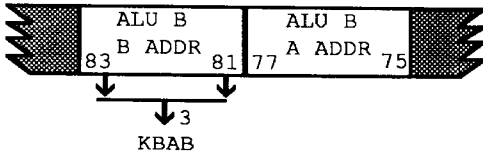


Bits 80:78 - ALU B Address A (*KBAA*)



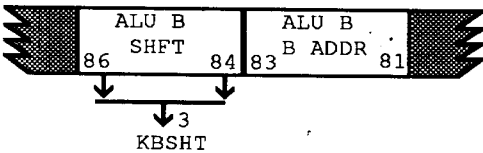
Provides the A register file address of ALU B. In the 32-bit mode, these bits are brought out to the I/O pads for user definition. (SU2:0)

Bits 83:81 - ALU B Address B (KBAB)



Provides the B register file address of ALU B. In the 32-bit mode, these bits are brought out to the I/O pads for user definition. (SU5:3)

Bits 86:84 - ALU B Shift and Destination Control (KBSHT)

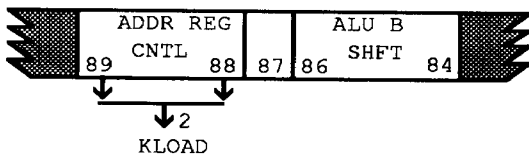


Provides the shift/destination instruction for ALU B. In the 32-bit mode, these bits are brought out to the I/O pads for user definition. (SU8:6)

Bit 87 - Word Swop (KWORD)

Commands the barrel rotator to rotate 16-bits on a word boundary.

Bit 89:88 - Address Register Control (KLOAD)



00	NOP
01	Load Address Register
10	Ld Address Reg and Data Page Reg
11	Enable Address Count

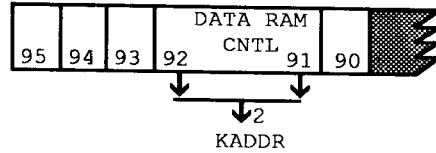
Bit 90 - Data RAM Select (KRAM)

Selects between ALU output and data bus as source for dcache input. The data bus source is useful when transferring FPU results to the data cache.

Bit 92:91 - Data RAM Control (KADDR)

00	Read dcache A & B
01	Write dcache A
10	Write dcache B
11	Write dcache A & B

Bit 93 - User Defined Bit (KUSR)



When using the 33xx FPU, this bit can provide the 34th control bit. Otherwise, can be used for any general purpose control in 32- or 16-bit mode.

Bit 94 - Control/Data Select (KDATA)

Decides whether the Shared Field contains immediate data or control.

Bit 95 - Byte Swop (KBYTE)

Commands the barrel rotator to rotate 8-bits on a byte boundary or when used in conjunction with **KWORD** provides a 24-bit rotate.

Development Tools

MDS offers a CHRIS EPU Development System based on MetaStep™ to support user software and hardware development. The software is supported on the IBM PC/AT, SUN, and VAX. A sample instruction set is provided with each Development System. The instruction set is assemble-like and allows the user to program close to the architecture of the CHRIS EPU.

Also included with each Development System is a Definition Processor for developing instruction sets.

© Copyright 1988 by MDS. All rights reserved.

CHRIS™ and CHRIS EPU™ are trademarks of Matra Design Semiconductor.
MetaStep™ and Definition Processor™ are trademarks of Step Engineering.

VAX is a trademark of Digital Equipment Corporation.
SUN is a trademark of Sun Microsystems, Inc.
IBM and PC/AT are trademarks of IBM Corporation.

Matra Design Semiconductor

2895 Northwestern Parkway
Santa Clara, CA 95051

12 TEL: 408/986-9000 TLX: 299-656 FAX: 408/748-1038

007097 _ _ _