Standard

MLP-900 MULTI-LINGUAL PROCESSOR

* * *

PRINCIPLES OF OPERATION

Standard
Standard Computer Corporation

FORM 809001-56

MLP-900 MULTI-LINGUAL PROCESSOR

* * *

PRINCIPLES OF OPERATION

RESTRICTED DISTRIBUTION

This document contains preliminary information on an unreleased product. Specifications are subject to change without notice. STANDARD Computer Corporation therefore assumes no liability.

RESTRICTED DISTRIBUTION

FORM 809001-6

Revised, May 1970

Copies of this and other STANDARD Computer Corporation publications can be obtained through the nearest STANDARD sales office (see back cover).

Comments concerning the contents of this publication should be addressed to STANDARD Computer Corporation, Technical Publications, 633 East Young St., Santa Ana, California 92705.

5-70-C-7C Printed in USA

PREFACE

This document is an introduction to the functional design of the MLP-900 Multi-Lingual Processor. It includes a description of system architecture, processing facilities and MLP-900 order codes.

The MLP-900 is a microprogrammed processor with an on-line alterable control memory. It possesses a designed-in capability for efficient interpretive execution of instructions in formats and languages other than its own internal order code set. On-line alterable microprogramming is not the only capability needed to make a processor efficient in multi-lingual interpretive applications. Three design principles were followed to develop a processor capable of handling a broad spectrum of target languages:

- The system architecture is generalized and provides direct control of logic functions and access to all storage elements.
- Order codes are well-structured and are similar in function to typical machine language instruction sets.
- The processor uses language-dependent hardware with complementary microprograms to efficiently adapt to widely varying target instruction and operand formats, I/O structures and memory addressing modes.

The language-dependent, plug-in hardware is used to perform a variety of translation and formatting functions more efficiently than unassisted software. Hardware aids can provide:

- High speed in executing repetitive functions; such as target instruction decomposition, order code translation and microprogram execution routine entry.
- Formatting of memory addresses and translation functions, such as page or byte addressing, memory protection, relocation, etc.
- Reduction in Control Memory storage requirements.

STANDARD Computer Corporation representatives are available to aid you in establishing requirements, specifications and performance goals for complementary firmware and hardware application packages for the MLP-900.

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

LIST OF ILLUSTRATIONS

LIST OF TABLES

SECTION 1
MLP-900 PROCESSOR DESIGN

1.1 GENERAL

The STANDARD MLP-900 Multi-Lingual Processor is a microprogrammed data processor designed to provide a high degree of versatility and computing power. One of the most important features is the high-speed, on-line alterable microprogram memory. The order code is designed for ease of use. The processor language is called MINIFLOW*; individual instructions are "ministeps". MINIFLOW programming resembles certain aspects of machine language programming on earlier processors. The MLP-900 is designed for direct access at the microprogram level. The MINIFLOW programmer can "get to" all mode control flip-flops and machine registers. MINIFLOW routines and complementary hardware tailor the MLP-900 for efficient execution of a broad spectrum of target languages and systems. High level procedures in existing languages can be executed in MINIFLOW directly. Complex macros and subroutines can be synthesized in MINIFLOW and execution can be initiated by the fetch of a single, problem-oriented, target language instruction.

In a typical processor, the instruction set generally can be separated by function into three categories:

a. Manipulation of arithmetic and logical operands.

b. Control and sequencing of instruction execution.

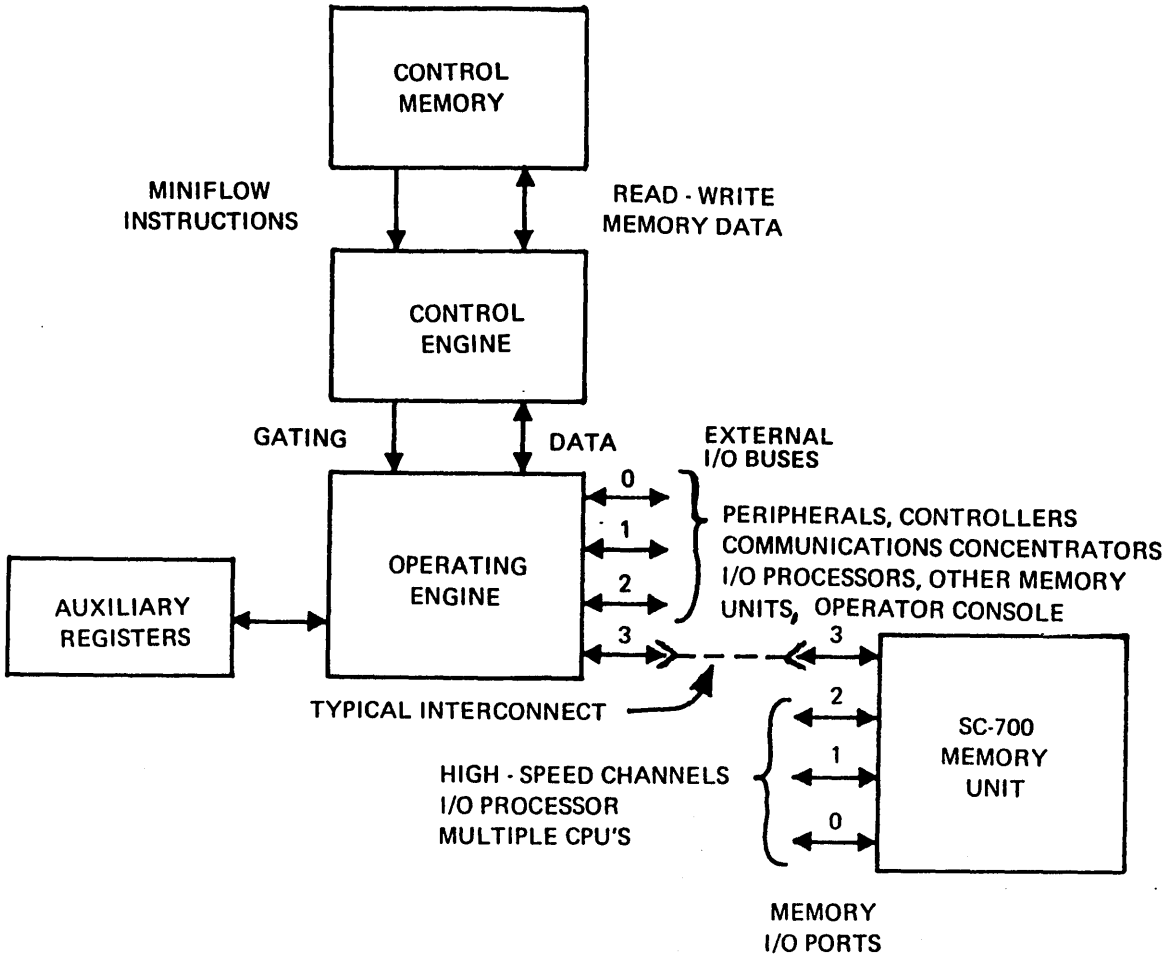
c. Input/output operations (which often combine both arithmetic and control functions).

In the STANDARD MLP-900, the facilities for performing arithmetic and logical transformations on data have been separated from the control and sequencing functions to a large degree. Control signal and data transfer interfaces provide communication between the data handling section, or Operating Engine, and the sequencer, or Control Engine. The Control Engine contains the microprogram (control) memory. Input/output operations, as in other processors, require considerable interaction between the sequencing and the data handling elements of the MLP-900. See Figure 1-1.

MINIFLOW instruction words are similarly divided into two types, called Operating and Control ministeps, which sequence the elements of the Operating and Control Engines, respectively. If an Operating ministep is followed in sequence by a Control ministep, the two will be executed simultaneously. Otherwise, a single Control ministep or one Operating ministep is executed.

Some processor tasks, which are highly repetitive in nature, are relatively inefficient when executed by an unassisted firmware routine. The MLP-900 has provision for up to four sets of Language Boards. The principal functions of Language Boards are to generate addresses and sequence accesses to Main Memory, extract data from target language instructions, and generate MINIFLOW entry addresses (branch vectors).

**MINIFLOW is a trademark of STANDARD Computer Corporation.*



NOTE: THE SC-700 MEMORY UNIT IS EXTERNAL TO THE MLP-900 PROCESSOR. ANY MLP-900 BUS CAN BE CONNECTED TO ANY SC-700 PORT. UP TO EIGHT 32K-WORD SC-700 MEMORY MODULES FIT INTO A SINGLE SC-700 CABINET. UP TO SIXTEEN CABINETS CAN BE ACCOMMODATED ON ONE BUS.

Figure 1-1. MLP-900 General Block Diagram

MINIFLOW language design and the MLP-900 hardware provide the following capabilities:

- On-line alterable microprogram (Read/Write Control Memory).
- Expandable Control Memory
- Interpretive execution of a wide range of target languages
- Multiple general-purpose registers

- Up to 1024 high-speed data registers (optional)
- BCD and byte arithmetic
- Multiple, independent, asynchronous, Input/Output buses
- Variable data field manipulation
- Direct access to all registers and control flip-flops
- Indirect register addressing at the microprogram level
- Flexible microprogram branching and transfer of control
- Microprogram subroutine entry and return
- Order code organized for efficient use of Control Memory

1.2 MLP-900 SUMMARY SPECIFICATIONS

- 128 nanosecond clock cycle
- 32 General Registers; 36 bits plus 4 parity
- Up to 1024 Auxiliary Registers*; 36 bits plus 4 parity
- Expandable Control Memory up to 64K (K = 1024) locations; 32 bits plus 1 parity
- Both Read-Write and Read-Only Control Memory available in 512-word modular increments
- 32 Data Mask Registers; 36 bits plus 4 parity
- Four independent, asynchronous, half-duplex, External (Input/Output) Buses; 36 bits plus 4 parity
- Automatic microprogram subroutine stacking to 15 levels
- Up to 4 target languages (independent order code sets) per processor

*Optional in increments of 256 words.

1.3 OPERATING ENGINE DESIGN

Figure 1-2 is the functional block diagram of the Operating Engine. This portion of the MLP-900 contains registers, shifters and arithmetic and logical elements used to manipulate operands. The Operating Engine External Buses interface with Main Memory, peripherals, and other external devices. Several interfaces provide communication with the Control Engine. The Operating Engine can execute seven Operating ministep types:

a. General Arithmetic (GEAR)--Performs binary arithmetic, logical operations, and single register shifts

b. Character/Left (CHAL) -- Byte and BCD operations from right to left

c. Character/Right (CHAR) -- Byte and BCD operations from left to right

d. Conditional External Data Exchange (CEDE) -- Transfers addresses, target instructions and data between the MLP-900 and Main Memory

e. Transfer External (TEXT) -- Transfers addresses and data between the MLP-900 and devices on the External Buses

f. Shift Instruction (SHIN) -- Executes complex shift operations

g. General Data Transfer (GENT) -- Transfers data between Operating Engine Registers and to and from the Control Engine interface

The nominal Operating Engine register length is 36 bits. The Primary Adder operates on two 36-bit inputs and the Byte/Decimal Adder manipulates two 8-bit bytes. Parity is maintained on 9-bit bytes during data transfers in the Operating Engine. Logical transformations which do not maintain parity, such as arithmetic and shifting operations, are performed by two identical, independent, logical structures. The outputs of the two elements are compared at clock time for identity. If they are identical, parity is regenerated; otherwise, an Action Request (microprogram interrupt) is forced. Byte parity (9-bit) is sent out over the External Buses and incoming parity is checked when data is moved from the External Bus Registers.

The Operating Engine accommodates up to four Language Boards. These provide a hardware assist to data exchange operations (CEDE ministeps) which process target language instructions and format Main Memory addresses. Each Language Board in the Operating Engine is matched with a corresponding Control Engine (CE) Language Board. Operating Engine (OE) Language Boards format addresses and commands to Main Memory. They also gate indirect addresses to General Registers, control the Primary Adder and transfer data to External Bus Registers. These functions allow

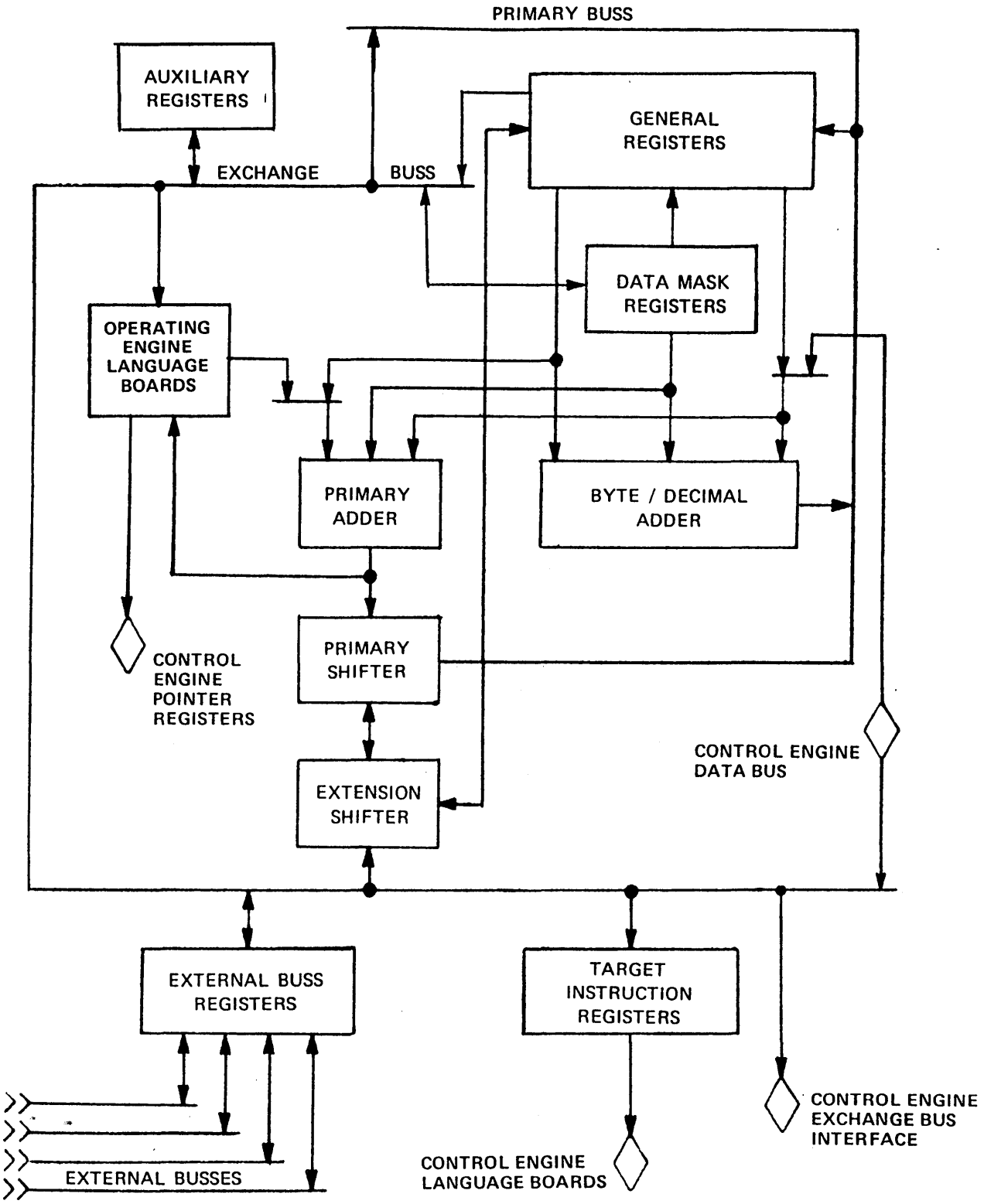


Figure 1-2. Operating Engine Block Diagram

certain types of fetches from memory to be addressed by data in an incoming word. The third major function of the OE Language Boards is to translate target language order codes into initial MINIFLOW entry addresses. Memory protection, certain types of indexing, page searching and other special-purpose address modification functions can be performed by OE Language Boards. Language Boards translate order codes and extract data and address fields faster than unaided MINIFLOW. Any function provided by either the OE or CE Language Boards can be accomplished by a MINIFLOW routine at the cost of processing speed and increased Control Memory space.

The MLP-900 accommodates up to 1024 full-word Auxiliary Registers. These registers may be used as high-speed (one-clock-time access) buffers, temporary data storage, etc. Auxiliary Registers are optionally available in increments of 256 words.

A data masking capability allows manipulation of fields shorter than the full Operating Engine register length. Thirty-two Data Mask Registers, 36 bits wide, are preloaded with mask words via MINIFLOW initialization routines. Mask words, or Masks, modify the operation of both the Primary and Byte/Decimal Adders and data transfers into the result register from the Primary Bus. One data masking mode allows loading a field into a register without changing masked-out bits. In the General Arithmetic (GEAR) ministep Clear mode, masked-out bits are zero-set in the result register. Only masked-in data is transferred in either mode. Arithmetic and logical operations apply to masked-in fields and ignore masked-out bits.

A number of register addresses other than General Registers are addressable during CEDE, TEXT and GENT execution. 1024 addresses are reserved for communicating with the OE Language Board Registers. Other addresses are reserved for the Target Instruction Registers (Primary and Secondary), the CE Data Buss, Data Mask Registers, Auxiliary Registers and External Registers.

Ministeps are obtained two at a time from Control Memory each clock cycle. Operating Ministeps are executed out of the OE Ministep Register (refer to Figure 1-4) in the Control Engine. Decoded control signals, from the OE Ministep Register, sequence the logical elements of the Operating Engine.

1.3.1 General Registers

The MLP-900 has 32 General Registers, each 36 data bits wide. Four parity bits, one for each 9-bit byte, are maintained with each register. All 32 registers are addressable as inputs to the Primary and Byte/Decimal Adders. Except for General Register 31, the Shift Extension Register, none of the General Registers has a dedicated function. General Registers have two independent address structures. The Operand A register specified by an Operating ministep is gated to one input of the Adders and the Operand B input goes to the other, depending on the addressing options of the ministep. Any of the General Registers may be specified as either, or both, Operand A or B inputs to the Adders by most Operating ministeps. Both Operand A and B registers can also be addressed indirectly. The indirect address mode of an Operating ministep uses the

five low-order bits in one of the 16 CE Pointer Registers (refer to paragraph 1.4.5) to specify the address of the corresponding operand in the General Registers.

Transfers of data to the General Registers from the Primary Bus (GEAR, CHAL, CHAR, and SHIN ministeps) are modified by the contents of one of 32 Data Mask Registers. In the Non-clear or replacement mode of data masking operations, logic prevents masked-out bits (corresponding to zeros in the mask word) from being changed in the General Registers. In the Clear mode of the GEAR ministep, masked-out bits are replaced with zeros. When the Test function of the GEAR, CHAL, CHAR, SHIN, and CEDE ministeps is enabled, the Operand A General Register is not loaded with the result of the operation, regardless of masking functions.

Odd-numbered General Registers are also bused to the Extension Shifter. During double-register-length shift operations (SHIN ministep), the Extension Shifter is paired with the Primary Shifter. Even-odd pairs of General Registers can be specified. General Register 31, the Shift Extension Register, can be paired with any General Register as an operand for double-length shifts. Refer to paragraph 2.2.3 for a description of the Shift Instruction, shift paths and functions.

1.3.2 Primary Adder

The Primary Adder is a 36-bit, parallel binary, arithmetic and logical processor. GEAR ministeps execute the 16 operations listed in Table 1-1. The result of operating on the Operand A and B inputs is placed in the Operand A register, except when the TEST bit is true. The Primary Adder is also used with several CEDE, TEXT, and SHIN ministep types, but operations are limited to binary addition and subtraction.

Operand B inputs to the Primary Adder include General Registers, Long and Short Immediate (literal) operands and the contents of any one of 16 Control Engine Pointer Registers. Operand A inputs are the General Registers or External Bus Input Registers. Inputs from theExternal Bus Registers are gated to the Primary Adder by Operating Engine Language Boards to speed up memory fetches during some types of CEDE ministeps. The result of this operation goes to the associated External Bus Output Register as an External Command Word.

Data Mask words affect the Primary Adder operation. Masking allows the MINIFLOW programmer to perform arithmetic, logical manipulation and testing on variable fields. Masking functions are:

- Adder outputs are forced to zero in masked-out bit positions.
- Arithmetic carries are not generated in masked-out bits.
- Arithmetic carries are propagated over masked-out bits.

Table 1-1. Primary Adder Operations

Arithmetic		Logical	
$A \leftarrow A+B$		$A \leftarrow A \cdot B$	(Logical AND)
$A \leftarrow A+\overline{B}+1$	(A-B, 2's complement)	$A \leftarrow A \cdot \overline{B}$	
$A \leftarrow \overline{A}+B+1$	(B-A, 2's complement)	$A \leftarrow \overline{A} \cdot B$	
$A \leftarrow A+B+COF1$	(Conditional carry-in)	$A \leftarrow A \cup B$	(Logical OR)
$A \leftarrow A+\overline{B}+COF1$		$A \leftarrow A \cup \overline{B}$	
$A \leftarrow \overline{A}+B+COF1$		$A \leftarrow \overline{A} \cup B$	
$A \leftarrow B$	(Clear and add)	$A \leftarrow A \mathrm{E} B$	(Exclusive OR)
$A \leftarrow \overline{B}$	(1's complement)	$A \leftarrow A \mathrm{E} \overline{B}$	(Compare)

"A" is Operand A. "B" is Operand B. "$\overline{A}$" is the 1's complement of A. "COF1" is the Carry-out State flip-flop used as initial carry-in. "$\cdot$" represents the logical product (AND) operation. "U" represents the logical union (OR) operation. "E" represents the Exclusive OR function. " $\leftarrow$ " (left arrow) denotes that the result of the operation on the right is transferred to the location specified on the left.

1.3.3 Byte/Decimal Adder

The MLP-900 Byte/Decimal Adder operates on 8-bit bytes. During the Character/Left (CHAL) and Character/Right (CHAR) ministeps, the Byte/Decimal Adder combines an 8-bit Operand A with an 8-bit Operand B input. The result goes to the Operand A location unless the TEST bit is on. Operand A inputs are General Registers. Operand B inputs are General Registers and auxiliary Operand B inputs (refer to paragraph 1.3.2). The byte location within a word is also specified. Figure 1-3 shows byte boundaries within a 36-bit register word.

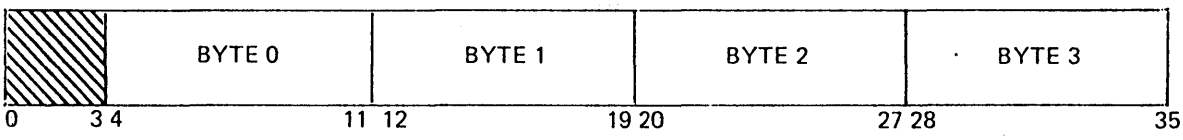


Figure 1-3. Byte Data Word Format

Four CE Pointer Registers (refer to paragraph 1.4.5) are available to tally byte and word locations in two operand character strings. Two of these four Pointers (P00 and P01) are used to indirectly address the A and B byte locations. The other two (P02 and P03) are used to hold character counters. For the CHAL ministeps, Pointers 00 and 02 are decremented if the A byte address is obtained indirectly. Pointers 01 and 03 are decremented if the B byte is indirectly addressed. Zero Sense and All Ones Sense State pseudo-flip-flop outputs, generated by the contents of the four Pointer Registers, are sampled to control sequencing. The CHAR ministep is similar. However, Pointers 00 and 01 are incremented instead of decremented when the associated byte address is indirect, and the pseudo flip-flops sampled for these two pointers would be "Three Sense" and "Four Sense". In both CHAL and CHAR if the TEST bit is true no incrementing or decrementing takes place even though byte address is specified indirect.

The Byte/Decimal Adder executes decimal operations on binary-coded-decimal (BCD) characters (two per byte) and binary operations on 8-bit bytes, as shown in Table 1-2. Decimal arithmetic combines the two-BCD-digit A and B bytes and generates a decimal sum or difference. Both inputs are checked for validity and the result is corrected for carries. Except when masked, binary operations affect the entire 8-bit operand field. CHAL and CHAR Masks are the eight least significant bits of the addressed Data Mask word.

Table 1-2. Byte/Decimal Adder Operations

Decimal	Binary		Logical
A ← A+B	A ← A+B	A ← A+B+COF1	A ← B
A ← A+$\hat{B}$	A ← A+$\overline{B}$+1	A ← A+$\overline{B}$+COF1	A ← A·B
A ← A+B+COF1	A ← $\overline{A}$+B+1	A ← $\overline{A}$+B+COF1	A ← AUB
A ← A+$\hat{B}$+COF1			A ← AEB

"$\hat{B}$" represents the 9's complement of two BCD characters in the B byte.

1.3.4 Data Mask Registers and Functions

The MLP-900 Processor has 32 Data Mask Registers, each 36 bits wide. Mask Registers are divided into two banks of 16 registers. Bank selection is controlled by the Bank Select State flip-flop (refer to paragraph 1.4.8). Data masking allows selective operation on bits and fields within a word. Masking is effective during GEAR, CHAD and SHIN ministeps. A Mask Register is addressed each execution of these ministeps. The eight, low-order bits (bits 28-35) in the selected Mask Register are active for CHAD. During SHIN, masking is effective on data in the Primary Shifter. Bit positions in the selected Mask Register which contain a one (1) are defined as masked-in bits. Bits containing a zero (0) are masked-out bits.

During GEAR, Masking can operate in one of two modes. In the Clear mode, all masked-out bits of the result word transferred into the Operand A register are replaced by zeros; masked-in bits are read into the register. If the CLEAR bit is off, masked-out bits inhibit alteration of the contents of corresponding bits of the receiving register. Masked-in bits will be loaded normally. The Clear mode is equivalent to the generation of the logical product of the Mask word and the result word. When the CLEAR bit is false (zero), masked-on bits are loaded into the result register without disturbing data in masked-off bit positions, which is equivalent to field replacement.

Data Masks also modify the operating modes of the Primary and Byte/Decimal Adders, as described in paragraphs 1.3.2 and 1.3.3, respectively. In all masked-out bit positions, the Adder output is forced to zero. This makes it possible to test the result of the operation on masked-in fields. In the Adders, mask logic enables the propagation of carries through masked-out bits and suppresses carry generation. Arithmetic operations are executed correctly on fields shorter than the full register

word (or full byte width during the CHAL and CHAR ministeps). If a carry-out signal is generated in a masked-in field, it will propagate through masked-out bits. If it encounters no masked-in bits, it will be sensed as a Carry-Out signal (COP), and will be loaded into the Carry-Out flip-flop (COF1) at clock time.

Zeros (0) are shifted into vacated masked-in bit positions of the result word when a shift is executed. In the Clear mode of the GEAR ministep, data is shifted into masked-out bit positions without being lost.

1.3.5 Shift Operations

The Operating Engine has two functionally identical Shifters. Both are 36-bits wide and have some shift paths in common. The Primary Shifter operates on the output of the Primary Adder. The Extension Shifter operates on data in odd-numbered General Registers and External Bus Registers. Up to 16 bits of cross-over data is passed when a connected shift is executed. GEAR ministeps control only single-register shifts by the Primary Shifter. Shift Instruction (SHIN) ministeps can execute both single- or double-register shifts. SHIN codes can command connected or independent double-length shifts, single-length shifts, circular shifts (double- or single-length), primitives for iterated divide and multiply algorithms, normalize shifts and indirect shifts. The SHIFT AMOUNT field of the GEAR, and SHIN ministeps can specify one-clock-cycle shifts of left or right 0, 1, 2, 4, 6, 8, 12, and 16 bits.

SHIN indirect shifts use the Shift Control Pointer Register (P07) in the Control Engine. The 6- and 12-bit shifts cannot be executed by indirection. When an indirect shift is executed, the four least significant bits and the Logical OR of the four most significant bits in 8-bit-wide Shift Control Pointer Register are sampled. Since the maximum single shift is 16 bits, a 16-bit shift is taken each time logic detects a count greater than 16; i.e., a one (1) somewhere in bits 0-3 of the Shift Control Pointer. On execution, the Shift Amount count is reduced by 16. If the four most significant bits are zero, the next lower bit which contains a one (1) controls the shift. If the Shift Control Pointer has a count of 11 (binary 1011), for example, and the indirect shift is repeated, shifts of 8, 2, and 1 bit would be executed consecutively. The count in the Pointer is reduced by the amount of the shift each time. By pairing a "branch" type Control ministep to test the SHD (Shift Done) State pseudo-flip-flop, with a SHIN ministep, a 1-clock-time repetitive loop can be formed to execute indirect shifts until the Pointer count goes to zero.

Multiply and Divide capability are provided by repeated execution of corresponding SHIN ministeps. The number of iterations is initially placed in the Shift Control Pointer Register. The count in the Shift Control Pointer is reduced by one each iteration and the loop is normally terminated as the count goes to zero. Normalize shifts are controlled by outputs from the CE Language Board. A tally is kept in the Shift Control Pointer during Normalize. An output of the Language Boards, the Normalize Done State pseudo-flip-flop, is available for testing to control the process. Specifications for SHIN ministep functions are located in paragraph 2.2.3.

1.3.6 External Buses and Registers

Except for some miscellaneous control outputs and interrupt inputs, the MLP-900 Processor I/O interface consists of up to four External Buses. Each is 36 data bits and 4 parity bits wide. Each Bus uses an associated group of control, signalling and timing lines to sequence data transfers in both directions. Buses are bidirectional, time shared (half-duplex) communications ports, designed to send addresses and data out and receive data back with equal facility. The MLP-900 communicates with devices on the bus by way of up to 8 External Bus Registers. Each Bus has a register for incoming data and a register for outgoing addresses and data. Transfers to and from the External Bus Registers are sequenced by Conditional External Data Exchange (CEDE) and Transfer External (TEXT) ministeps. CEDE ministeps are designed for fetching target instructions and operands from Main Memory. TEXT is used to communicate with peripherals. Various fields of incoming target language instructions are processed by the MLP-900 Language Boards under the control of the CEDE ministep. Some of the CEDE ministeps provide semi-automatic sequencing of instruction and operand fetches. A number of CEDE and TEXT varieties place the processor into the Wait mode if an external device has not responded to a request when execution begins. During waits, ministep execution is inhibited until the desired operation occurs, or an Action Request forces a transfer of control.

The MLP-900 Processor samples four parity bits on each input transfer. If bad parity is detected during a fetch sequence, the previously transmitted external address word in the Output Register allows a retry of the fetch. This greatly improves the possibility of recovering from a transient error on the input.

The MLP-900 is designed to work with the SC-700 Magnetic Core memory. With proper interfacing, however, the processor can be adapted to communicate with almost any type of self-clocked memory unit, since data transfers inward have few timing constraints. Appendix A of this document is a short description of the SC-700 memory.

1.3.7 Operating Engine Language Boards

Operating Engine (OE) Language Boards perform a wide variety of tasks related to the generation of MINIFLOW entry addresses from target language instructions and formatting address (External Command) words to Main Memory. In conjunction with a corresponding Control Engine (CE) Language Board, an OE Language Board adapts the MLP-900 to a specific target language (instruction set) and target environment (system framework). Up to four pairs of Language Boards can be installed in each MLP-900 Processor. Appendix B contains a more detailed description of the MLP-900 OE and CE Language Board inputs, outputs and typical functions.

OE Language Boards have two primary functions. One is address modification for accesses to Main Memory. Address modification is effective when External Command words, generated by CEDE ministeps, initiate fetches of target language instructions and operands and generate write addresses. Address modification facilities

translate target system addresses into equivalent Main Memory addresses. Typical functions include:

- Insert Memory Protect keys
- Modify memory addresses using base register inputs
- Perform address limit violation checks
- Translate relocated page addresses
- Translate character and byte addresses to word addresses

Another major function of the OE Language Boards is to generate hard-wired, initial entry addresses to initialization MINIFLOW during execution of the CEDE/WIN (Wait for Instruction) ministep, based on target instruction type. Initialization MINIFLOW routines include such functions as operand fetching, indexing, indirection, and some complex types of address modification. After the initialization routine is completed, the execution MINIFLOW routine is entered by branching with the assistance of the Control Engine Language Boards.

Several secondary functions can be performed by the OE Language boards, such as generating operand addresses from target instruction data. Similarly, OE Language Boards can execute fetches, using incoming data, during indirect addressing modes. Various CEDE ministep types are implemented to provide semi-automatic addressing sequences.

1.3.8 Target Instruction Registers

There are two Instruction Registers, Primary Instruction Register and Secondary Instruction Register, both of which can be loaded from the active External In Register as controlled by Operating Engine Language Boards. The Operating Engine Language Boards can also cause the Primary Instruction Register to be loaded from the Secondary Instruction Register. The setting of these registers would normally be accomplished during a CEDE/WIN. They are Operating Engine registers and may be addressed in the usual fashion by a GENT or CEDE ministep.

The Target Instruction Registers are sensible by the CE Language Boards. This allows target instruction field extraction.

1.4 CONTROL ENGINE DESIGN

Figure 1-4 is a functional block diagram of the MLP-900 Control Engine. The Control Engine can be conceptually divided into two smaller functional groups. One is made up of the Control Memory Address Generators, the Control Memory itself, its Output Registers and the Operating and the Control Engine Gating structures. The second group sequences data transfers, stores data, monitors status and controls testing and sequencing. Major operations of this part of the Control Engine include:

- Generate Control Memory address inputs
- Translate target language instructions and extract data fields
- Provide iteration control and indirect register addressing
- Hold "Status" data and tally external and internal interrupts.
- Interface between the Operating and Control Engine registers.

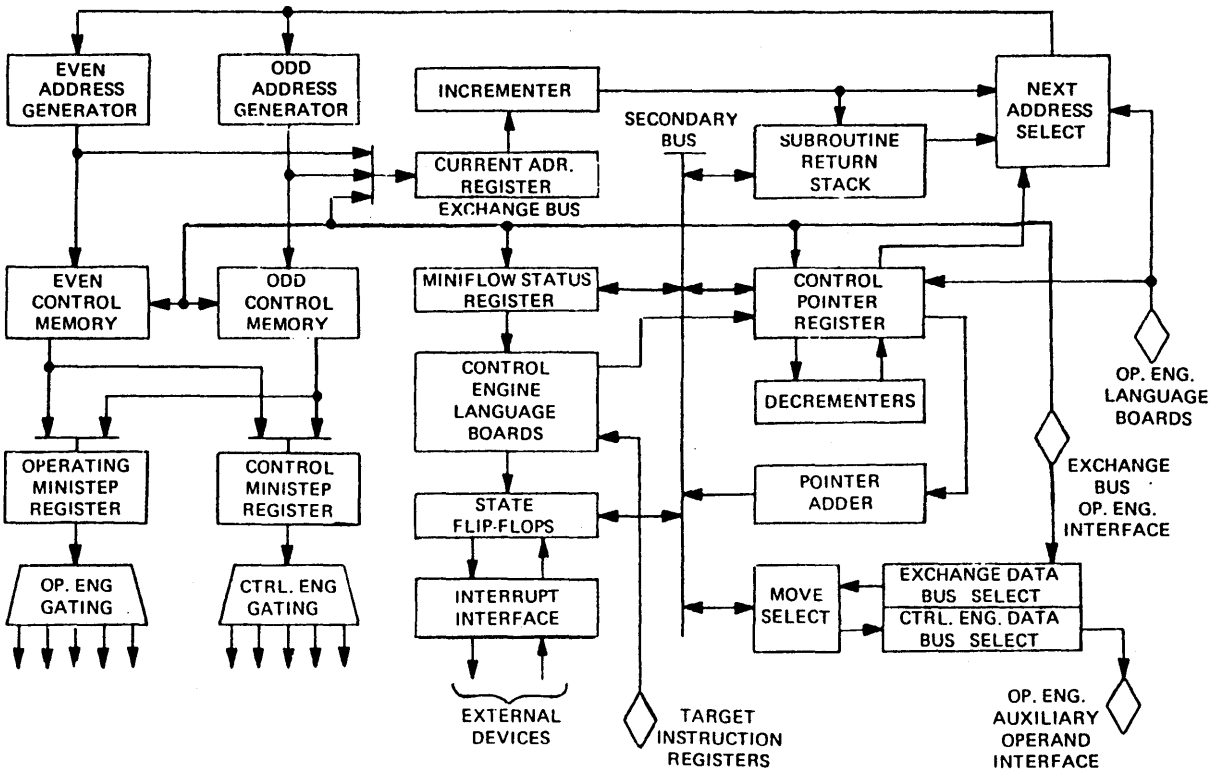


Figure 1-4. Control Engine Block Diagram

Two sequential ministeps are fetched each clock cycle. The even and odd Control Memory banks are completely independent. Current ministep addresses may be either even or odd without restriction. Two ministeps will be executed simultaneously when an Operating ministep is followed sequentially by a Control ministep (with one exception). Only the current ministep is executed for other sequences. Control Engine ministeps take no additional processing time when paired. It is not necessary to pair ministeps if there are no processing benefits. This may improve efficiency by saving some locations in Control Memory.

Control ministeps sequence Control Engine logic just as Operating ministeps regulate Operating Engine functions. Eight Control ministep varieties are available. There are five Branch ministeps, a Block Transfer ministep for data transfers between Control Memory or the Subroutine Return Registers and the Operating Engine interface, a ministep for manipulating the State (mode control) flip-flops and a ministep to move data between Control Memory registers.

Branch ministeps test State flip-flops which log the processor current status. Branch ministeps can be used to modify the sequence of Control Memory accesses if the branch test condition is true. Branch ministep execution can alter the sequential execution of the microprogram instruction in several ways. Least complex is the Branch Test (BRAT) ministep, which executes relative branches based on testing one or a combination of two State flip-flops. Branch and Enter (BENT) is similar to BRAT, except taking the branch causes the return address to be loaded into the Subroutine Return Stack. The Branch Or Return (BORE) ministep executes an automatic subroutine return if the branch is not taken. Branch and Modify (BRAD) modifies the amount contained in one of eight Control Engine Pointer Registers by the count specified in the ministep. One of 256 State flip-flops is tested by BRAD.

The Branch-Extended Address (BEAD) ministep has four modes of execution. Two modes execute absolute branches, indexed and unindexed, to any location in the Control Memory. When the branch is made unindexed in one mode, a State flip-flop is tested. The indexed absolute branch adds the contents of one of the Pointer Registers to the 16-bit Extended Branch Address field in the ministep. The third mode allows conditional branches with relative addressing to the full extent of control memory. The fourth mode adds the contents of a Control Engine Pointer Register to the continuation address and transfers. The BEAD ministep can execute a subroutine entry in all modes.

There are three Control ministeps other than Branches. The Manipulate Status (MAST) ministep operates directly on State flip-flops. Any two of the State flip-flops are addressed independently and a combination of their logical states sets or resets the result flip-flop. MAST cannot affect State pseudo-flip-flops (refer to paragraph 1.4.8). Block Transfer (BLOT) loads and reads out the contents of Control Memory and the Control Engine Subroutine Stack (refer to paragraph 1.4.4) and sequences Operating Engine data transfers. BLOT is usually paired with an Operating Engine CEDE, TEXT, or GENT ministep. The BLOT ministep uses "Load Control Words" to execute chained (scatter/load) transfers into the processor. The Control Engine MOVE ministep transfers data between Control Engine Registers and, in conjunction with the GENT, CEDE,

and TEXT ministeps, to and from the Operating Engine. Data is transferred in 8- or 16-bit bytes, depending on the MOVE mode selected. The MOVE ministep also has a mask capability on 8-bit transfers.

1.4.1 Control Memory Addressing

Control Memory address inputs are gated from a number of sources to the Control Memory Address Generators by the Next Address Select logic. The major inputs are from the incremented Current Address Register (when no branching or transfer of control takes place), Control Engine Pointer Registers, the Subroutine Return Stack (when a subroutine is EXITED), the Branch Address fields of various Control ministeps, and the OE Language Boards when a CEDE/WIN (Wait for Instruction) Operating ministep is executed. The CEDE/WIN Ministep generates a starting address for MINIFLOW routines to initialize target language instruction processing. Refer to Figure 1-5.

Another input, not shown in Figure 1-2, is a special-purpose Entry Address Generator, which provides hard-wired transfer addresses into error and interrupt MINIFLOW routines. Also not shown are inputs from Maintenance Console switches which are used for trouble-shooting and initialization.

The Even and Odd Address Generators are adders and incrementers which form Control Memory bank addresses from the base address inputs and modifiers gated by the Next Address Select logic. Several modes of addressing are synthesized from these inputs. Departing from the nominally sequential, i.e., current address plus increment (continuation address) and current address plus increment plus one; we have (neglecting the incremented address of the second ministep):

- Branch Relative
- Branch Absolute
- Branch Absolute plus Pointer Count
- Branch and Store Subroutine Return
- Branch and Modify Pointer Count
- Continuation plus Pointer Count
- Subroutine Return
- "Forced" Transfers

High-priority forced transfers occur immediately when internal monitoring logic detects an error Action Request interrupt condition, Lower-priority Action Requests take effect during CEDE Wait modes. Any of a number of lower priority Target System Interrupts can cause a forced transfer if present when a CEDE/WIN ministep execution is attempted.

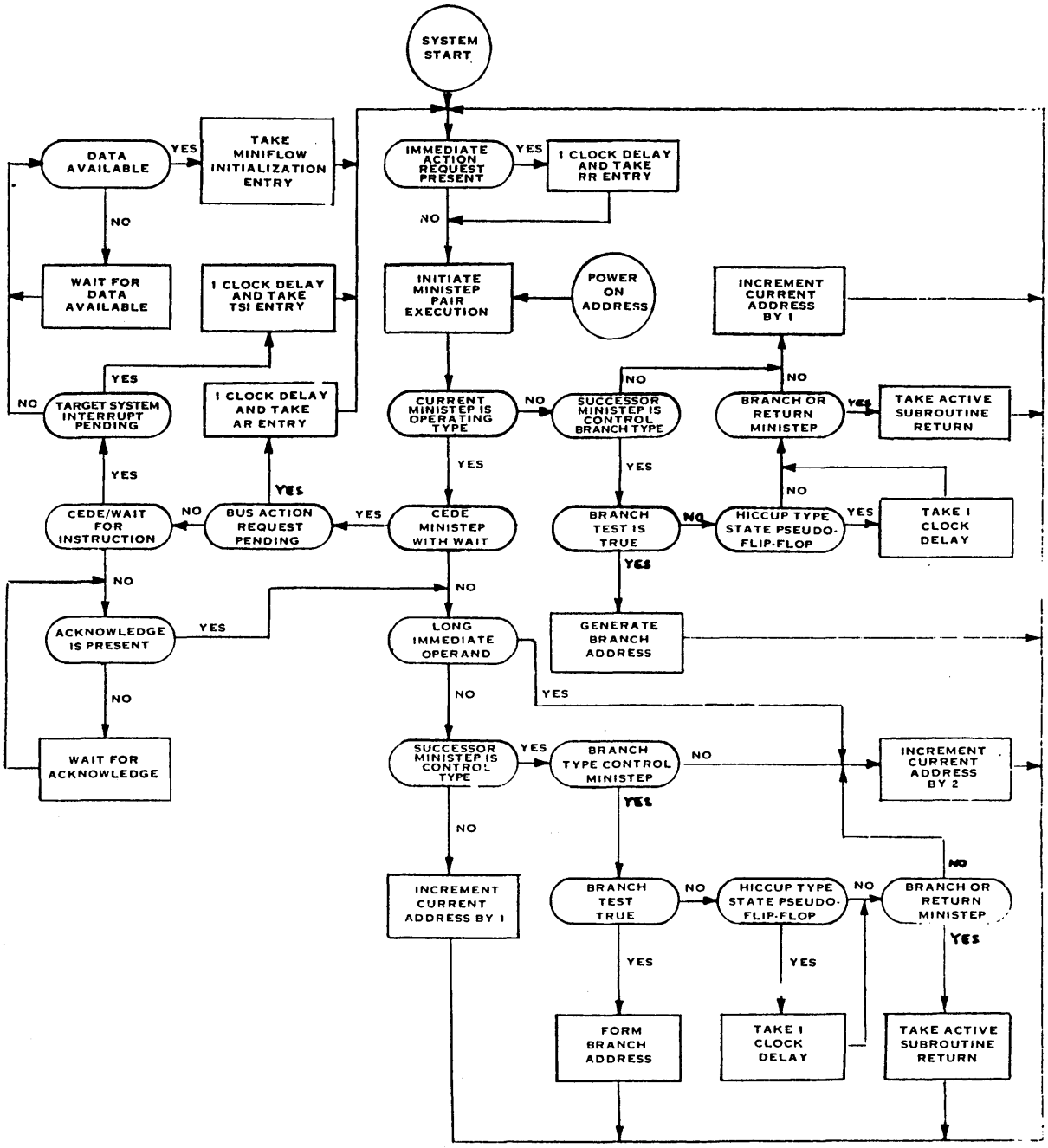


Figure 1-5. Control Memory Address Selection, Flow Diagram

1.4.2 Control Memory Design

The Control Memory is organized into odd and even banks. Two sequential ministeps are accessed at a time. An important feature of the MLP-900 Control Memory is that it is available in either Read-Write or Read-Only versions. The high-speed, Read-Write Storage (RWS) of the MLP-900 lets the user modify microprograms practically at will. The MLP-900 Processor with a RWS is highly adaptable and can peform processing functions in different languages (order codes) by reading in a new MINIFLOW and enabling the corresponding set of Language Boards.

Read-Write Store modules are composed of 512 32-bit (plus parity), semiconductor registers. To change the microprogram during operation and for initialization when the processor is turned on, the RWS is loaded from an external storage element. The BLOT (Block Transfer) ministep is used for initializing and overwriting the RWS. Due to its flexibility, RWS is highly useful for program check-out, debugging, maintenance and sequential overlay of MINIFLOW routines.

Read-Only Store (ROS) is a high-speed, 32-bit word (plus parity) memory with contents which are permanently fixed when the module is built. Data in the ROS cannot be altered by over-write, power loss or transient conditions. The contents of ROS modules must be completely specified at the time hardware is ordered. ROS modules currently contain 128 words (64 even and 64 odd locations).

1.4.3 Ministep Registers and Gating Functions

Even Control Memory and Odd Control Memory outputs are read into the Operating Ministep Register and Control Ministep Register, depending on sequencing and ministep type. After the Ministep Registers are loaded, their contents are further decoded and passed to gating structures which sequence the Operating and Control Engine elements. After each access of the Control Memory, the Operating and Control Ministep Registers hold the currently addressed MINIFLOW word and its next sequential successor ministep. The current ministep is read from the Control Memory location which has the numerically smaller address of the pair. It may be located at either an odd or an even address. The most significant bit in the OP CODE field (bit zero) determines whether the current ministep is routed to the Operating or Control Engine Gating structure.

If the current ministep is an Operating type and its successor is a Control type, both ministeps will be executed simultaneously with these exceptions:

- If the current ministep is an Operating type and it specifies a Long Immediate Data word (literal operand), the successor word is not an instruction. From the register, it is gated to the Control Engine Data Bus. From there it goes to the Operating Engine as an auxiliary Operand B input of the Primary Adder.

- CEDE/WIN ministeps inhibit execution of any successor ministep, since a forced branch is always taken (to an even Control Memory location).

It is possible to have a Control ministep for the current ministep. When this happens, the current ministep has priority of execution over the successor ministep. The current ministep is routed to the Control Engine Gating when it is a Control ministep and the successor ministep is not executed. Likewise, if the two sequential ministeps are both Operating ministeps, the current ministep has a priority of execution and it is routed to the Operating Engine Gating elements. The successor ministep is not executed at this clock. If no branch is taken, the Current Address Register is augmented by one and the successor ministep and its successor are read from Control Memory the next cycle.

1.4.4 Subroutine Return Stack and Stack Control

The Subroutine Return Stack is a group of 16 storage registers which hold 16-bit return addresses. A subroutine Return Register is loaded each time a subroutine entry is executed. Entries are made when a BEAD (Branch-Extended Address), with the ENTRY bit on, or a BENT (Branch and Enter) ministep executes a branch. Entries are also made when a breakout from the normal MINIFLOW sequence is forced by an Action Request (interrupt). Destacking occurs on a BORE (Branch Or Return) ministep that does not take the branch.

Loading of subroutine return addresses is scheduled by the Stack Pointer Register which addresses the top of the stack (the active return address). Pointer Register 06 is dedicated to this function. When the four least significant bits of the Stack Pointer are zero, the stack is empty. Initial entry into an empty stack is location 01. Subsequent entries go into consecutive ascending locations. A Stack Full Action Request occurs when location 15 is loaded. At the next clock, the now current address is entered in Stack Register 00 and a forced transfer to the Stack Full service routine is executed. A Stack Underflow Action Request occurs when a return is executed and the four least significant bits of the Stack Pointer Register are zero (0), and Action Request Lockout ARL2 is off. The Stack Pointer will decrement, and at the next clock the Action Request breakout will enter the address of the return ministep in Stack Register 00, and control will be transferred to the Stack Underflow service routine.

Although the usable Subroutine Return stack capacity is only 15 words, with an 8-bit Stack Pointer and two MINIFLOW service routines, the stack capacity can be easily extended to 240 levels (15 x 16). More complex service routines can extend the stack depth indefinitely or handle multiple stacks. However, the stack full service routine must detect a real upper-bound stack overflow error, otherwise a programming error can cause subroutine entry to an infinite number of levels. Likewise, the underflow MINIFLOW routine must check for a real lower-bound stack limit for the same reason.

1.4.5 Pointer Registers

A group of 16 Control Engine Pointer Registers are available for counting and indirect addressing functions. Register locations 00-07 are counting Pointers. Locations 08-15 are pseudo-register (non-counting) Pointers, mechanized as sense line outputs, with functions determined by Language Boards. Most of the counting Pointers

perform special services for other Control and Operating Engine logic elements at various times. Otherwise they can be used for general processing functions. Table 1-3 is a list of Pointer assignments.

Table 1-3. Pointer Register Functions

Pointer Register	Function
00 - 03	CHAL, CHAR, BLOT sequence Counters
04,05	General Purpose (not dedicated)
06	Subroutine Stack Pointer
07	Shift Control Register
08 - 14	CE Language Boards (Pseudo-register)
15	OE Language Boards (Pseudo-register)

Pointer Register contents indirectly address General Registers for Operating ministeps which allow indirection on Operand A and B selection. In addition, the CHAL (Character/Left) and CHAR (Character/Right) ministeps (paragraph 2.2.4) allow indirect selection of the Byte A and Byte B Operands within the Operand A and B words. However, the CHAL and CHAR Byte A indirect address is specified by Pointer Register 00. The Byte B indirect address is obtained from Pointer 01. Pointers 02 and 03 are normally used as character string length counters for the A and B byte operands respectively when indirect addressing occurs. These four Pointer Registers have individual decrementers. In addition, Pointers 00 and 01 have individual incrementers. The count in the two pairs of Pointers is adjusted by one (1) during the execution of the CHAL and CHAR ministeps when the corresponding indirect byte address mode is specified. In addition to assisting the CHAL and CHAR ministeps, Pointers 00 - 03 are also used to automatically tally locations and word counts during execution of the BLOT (Block Transfer) ministep (refer to paragraph 2.3.6).

Pointer Registers 04 and 05 are undedicated registers and can be used for general-purpose counting or indirect addressing. Pointer 06 is the dedicated Subroutine Stack Pointer (described in paragraph 1.4.4). Pointer 07 holds and tallies indirect shift amounts for several of the Shift Instruction (SHIN) ministep types and maintains a shift count during SHIN Normalize (refer to paragraph 2.2.3). Pointer Pseudo-registers 08-15 represent data translated from target instructions by the Operating and Control Engine Language Boards. Pointers 08-14 are driven by CE Language Boards from the Target Instruction Registers in the Operating Engine. Pointer Pseudo-register 15 is similarly driven by the OE Language Boards. Pointer Pseudo-registers can be used for indirect addressing and auxiliary Operand B inputs, but their contents must be moved to one of the counting Pointer Registers for loop control, shift operations and similar functions.

In addition to the four dedicated single-count decrementers of Pointer Registers 00-03, and incrementers of Pointer Registers 00 and 01, a separate Pointer Adder can be used to alter the contents of the counting Pointer specified in a BRAD (Branch and Modify)

ministep by the amount in the MODIFIER field. The Pointer count can be changed by any amount from plus seven (+7) to minus eight (-8) in one clock cycle.

Most Pointer Registers have a State pseudo-flip-flop (ZSI00-11) which goes to the one (1) logic state when the corresponding Pointer count is zero (refer to paragraph 1.4.8). These Zero Sense State flip-flop indicators can be tested for zero (0) logic conditions during the counting process, to maintain loop control. In addition, Pointer Registers 00-03 are mechanized with All Ones Sense State pseudo-flip-flops (OSI00-03) which are true when the corresponding Pointer contains all ones. In addition, Pointer Registers 00 and 01 have pseudo-flip-flops indicating when they have the value of either three or four (Three Sense TSI00, TSI01 and Four Sense FSI00, FSI01). These outputs are normally used for loop control during CHAL and CHAR ministep execution.

Since the BRAD ministep can increment any counting Pointer (P00-P07) by an amount other than one, it is possible to pass through a zero count from either direction. Consequently, a test of the Pointer Zero Sense pseudo-flip-flop will fail. When the Pointer Register count goes through zero, the overflow (or underflow) output of the Pointer Adder is used to drive the Through Zero State pseudo-flip-flop. This condition is true only during the clock cycle in which the overflow or underflow occurs. Through Zero must be sampled during the execution of BRAD to be valid. Since this output is developed relatively late in the clock cycle, a false branch condition resulting from a test of Through Zero causes a clock inhibit and a "hiccup" or one-clock-cycle delay. When testing a Zero Sense or One Sense pseudo-flip-flop of one of the Pointers, the indication is present at the start of the next clock cycle after the counting operation is complete and testing cannot cause a "hiccup."

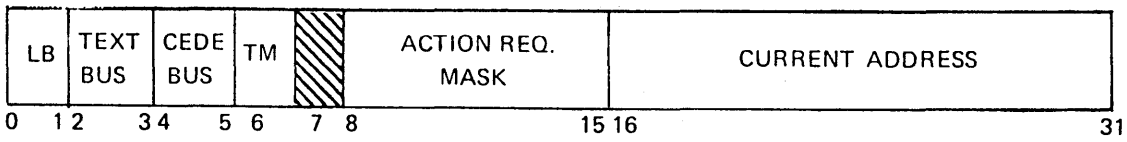


Figure 1-6. MINIFLOW Status Word Format

1.4.6 MINIFLOW Status Word

The two elements labeled in Figure 1-4 as "MINIFLOW Status Register" and "Current Address Register" are treated as a single 32-bit register by the BLOT (Block Transfer) ministep. The composite register holds the MINIFLOW Status Word (MSW) in the format shown in Figure 1-6. The MINIFLOW Status Word facilitates initializing the processor from a cold start and when a target language changeover is executed.

The Language Board Select (LB) bits determine which Language Board set is active when executing target instructions. These two bits enable one Control and Operating Engine Language Board pair (out of a possible four pairs). Two groups, of two bits each, separately select the External Bus address for I/O (TEXT BUS) and Memory (CEDE

BUS) communications. When the MSW Test Mode (TM) bit is on, the MLP-900 is in an abnormal state. This mode allows special operations to be performed to exercise and test automatic check and error Action Request logic. The true state (1) of the Test Mode bit enables:

- Writing into Control Memory with bad parity (BLOT/WBP ministep; paragraph 2.3.6).
- Reading Control Memory with bad parity for more than two consecutive clock cycles.
- Loading Operating Engine registers with bad parity (GENT ministep;paragraph 2.2.5).
- Forcing check errors (Check Test State flip-flop;paragraph 1.4.8).
- Setting the State flip-flop Clock Control (CKC) to zero (Machine Halt).

Action Request Mask (ACTION REQ. MASK) bits inhibit some Action Requests (AR's) or groups of AR's so that they cannot force a transfer of control (refer to paragraph 1.4.9).

A convenient way to initialize the MINIFLOW when executing a multiple block transfer (BLOT ministep) from an external source, is to load the MSW as the final step. After loading the MSW, the Current Address Register holds the start-up address for the new MINIFLOW.

1.4.7 Control Engine Language Boards

Four Control Engine (CE) Language Boards are paired with corresponding Operating Engine (OE) Language Boards. Language Board sets are selected by two bits in the Miniflow Status Word. The principle function of CE Language Boards to to extract and translate data from fields in target language instruction words. The mechanism for performing this function includes the Operating Engine Primary and Secondary Target Instruction Registers (refer to paragraph 1.3.8) which hold the data in original format; the CE Language Boards, which perform the decomposition and translation operations; and the Control Engine Pointer Pseudo-registers (refer to paragraph 1.4.5) and the CE Language Board State pseudo-flip-flops (refer to paragraph 1.4.8) which make the outputs available to the Control Engine. Appendix B contains a more detailed description of Language Board capabilities and interfaces.

CE Language Boards are passive and contain no internal storage elements. Extracted and translated data is selectively routed to CE Pointers 08-14, which can be tested, transferred or used as addresses or operands. If counting capability is required, the contents of Pointer Pseudo-registers must be moved to one of the countable Pointers. CE Language Boards develop the Group Entry address into the execution MINIFLOW from the target language instruction order code and place it in a Pointer Pseudo-register. The Group Entry address is called the "target entry branch table. Other fields which are normally held in Pointer Pseudo-registers include operand and index register addresses, character and word counts, shift amount data, literal operands (up to 8 bits), etc.

Target language instructions generally contain mode control data needed by the MINIFLOW programmer. To provide access to execution mode data, a group of State

pseudo-flip-flops (CLB00-11) are driven by the CE Language Boards. Control ministeps can sample these outputs to control execution sequencing. The Language Board Control State flip-flops (LBC 00-15) can be sensed by CE and OE Language Boards. They can be used to provide program control inputs to Language Boards as required.

Another function of the CE Language Boards is to examine the mantissa of floating point operands for leading zeros during execution of the SHIN/NORMALIZE Ministep. The output of this sampling process is the Shift Amount control to the Primary and Extension Shifters and a count (modulo the target exponent base) to tally the amount of normalization that takes place. An output is also generated when the process is complete (NMD--Normalize Done pseudo-flip-flop).

1.4.8 State Flip-flops

The Control Engine has a group of 256 addressable storage elements, called State flip-flops, which perform mode control functions. In most cases, State flip-flops are actual binary storage elements. Some are general-purpose and are available for scratch-pad use. Others are dedicated and are controlled by internal and external events. The remainder are synthesized from outputs of various logical elements in the processor and are called State pseudo-flip-flops. Table 1-4 is a listing of MLP-900 State flip-flops. They are grouped by characteristics and the addressing structure of the MOVE ministep which can access them. Appendix C describes the individual State flip-flops and their functions. Some general characteristics of the State flip-flops are as follows:

- General Indicator flip-flops are not dedicated and may be used as scratch-pad registers for housekeeping and status data, program monitoring and testing.
- Language Board (LB) Control (LBC00-15) flip-flops are available to use as mode control inputs to the OE and CE Language Boards. When they are not needed for this function, they may be used as general indicators.
- The External Write (EWR0-7) group can be mechanized to provide signals to external devices, separately from the External Buses. When not dedicated, they can be used as general indicators.
- The Action Request (AR) flip-flops respond to conditions which are to the MLP-900 Processor what system interrupts are to a machine language processor. They allow the normal MINIFLOW processing sequence to be modified by various internal conditions and external signals. When the hardware logic and the MINIFLOW simultaneously attempt to set and reset one of these flip-flops, the set input will dominate. The complement of Action Requests and a summary of their characteristics are described in paragraph 1.4.9.
- Control flip-flops are indicators which monitor status and control several important internal processor functions. When the hardware logic and the MINIFLOW simultaneously attempt to set and reset one of these flip-flops, the set input will dominate.
- Target System Interrupt flip-flops comprise four groups of eight bits each. For each Target System Interrupt flip-flop, there is a corresponding Interrupt Mask flip-flop. Each Interrupt Mask inhibits the true state of its corresponding Target System Interrupt flip-flop from causing a forced transfer of control to location 62 immediately prior to an attempt to execute a CEDE/WIN ministep.

Table 1-4. State Flip-Flop Listing
(Group 0)

General Indicator F/F's	General Indicator F/F's	Action Request F/F's		Control F/F's
Gen Ind GI00	LB Control LBC00	Power On	PON	Carry Out 1 COF1
Gen Ind GI01	LB Control LBC01	Power Off	POF	Carry Out 2 COF2
Gen Ind GI02	LB Control LBC02	Odd CM Parity	OPAR	Zero Flag 1 ZRF1
Gen Ind GI03	LB Control LBC03	Even CM Parity	EPAR	Zero Flag 2 ZRF2
Gen Ind GI04	LB Control LBC04	Invalid CM Addr	ICAD	Invalid Digit IDF
Gen Ind GI05	LB Control LBC05	Stack Full	STAF	Invalid Sign ISF
Gen Ind GI06	LB Control LBC06	Stack Underflow	STUF	Shift Out Sign SOS
Gen Ind GI07	LB Control LBC07	MINIFLOW Trace	TRAC	Shift Out Flag SOF
R00	R04	R08		R12
Gen Ind GI10	LB Control LBC10	Pri-Shift Error	PER	Last Bus In LBI
Gen Ind GI11	LB Control LBC11	Ext-Shift Error	XER	Shift Extnsn SHE
Gen Ind GI12	LB Control LBC12	A Bus Parity	APAR	Mask Bank Sel MBS
Gen Ind GI13	LB Control LBC13	B Bus Parity	BPAR	Mem Bus Inhibit MBI
Gen Ind GI14	LB Control LBC14	Extension Parity	XPAR	LB Indicators LI0
Gen Ind GI15	LB Control LBC15	Mask Parity	MPAR	LB Indicators LI1
Gen Ind GI16	LB Control LBC16	Exchange Bus Parity	ECP	LB Indicators LI2
Gen Ind GI17	LB Control LBC17	Ext Bus Parity	EBP	LB Indicators LI3
R01	R05	R09		R13
Gen Ind GI20	Gen Ind GI60	Bad Addr Bus 0	BAB0	TSI Inhibit SII
Gen Ind GI21	Gen Ind GI61	Bad Addr Bus 1	BAB1	TSI Lockout SIL
Gen Ind GI22	Gen Ind GI62	Bad Addr Bus 2	BAB2	Mem Cmnd 0 MC0
Gen Ind GI23	Gen Ind GI63	Bad Addr Bus 3	BAB3	Mem Cmnd 1 MC1
Gen Ind GI24	Gen Ind GI64	Language Board	LBR	Clock Control CKC
Gen Ind GI25	Gen Ind GI65	MM Addr Compare	ADC	Run F/F RUN
Gen Ind GI26	Gen Ind GI66	Ext Call Bus 0	CAB0	Check Test CKT
Gen Ind GI27	Gen Ind GI67	Ext Call Bus 1	CAB1	Initiate Trace ITR
R02	R06	R10		R14
Gen Ind GI30	Ext Write EWR0	Ext Call Bus 2	CAB2	AR Lockout 1 ARL1
Gen Ind GI31	Ext Write EWR1	Ext Call Bus 3	CAB3	AR Lockout 2 ARL2
Gen Ind GI32	Ext Write EWR2	Ext Interrupt 1	RUP1	AR Lockout 3 ARL3
Gen Ind GI33	Ext Write EWR3	Ext Interrupt 2	RUP2	AR Lockout 4 ARL4
Gen Ind GI34	Ext Write EWR4	Ext Interrupt 3	RUP3	AR Lockout 5 ARL5
Gen Ind GI35	Ext Write EWR5	Ext Interrupt 4	RUP4	
Gen Ind GI36	Ext Write EWR6	Ext Interrupt 5	RUP5	Retry Cntr 0 RTC0
Gen Ind GI37	Ext Write EWR7	Ext Interrupt 6	RUP6	Retry Cntr 1 RTC1
R03	R07	R11		R15

Table 1-4. State Flip-Flop Listing (continued)
(Group 1)

Target System Interrupts	Target System Interrupt Masks	State Pseudo-F/F's	State Pseudo-F/F's
Interrupt SI00	Intrpt Mask, IM00	Carry Out COP	Sense SW 0 SSW0
Interrupt SI01	Intrpt Mask, IM01	Zero Sense ZSP	Sense SW 1 SSW1
Interrupt SI02	Intrpt Mask, IM02	Invalid Digit IDP	Sense SW 2 SSW2
Interrupt SI03	Intrpt Mask, IM03	Invalid Sign ISP	Sense SW 3 SSW3
Interrupt SI04	Intrpt Mask, IM04	Thru Zero THZ	Sense SW 4 SSW4
Interrupt SI05	Intrpt Mask, IM05	Wait AR Pending WAR	Sense SW 5 SSW5
Interrupt SI06	Intrpt Mask, IM06	Normalize Done NMD	Sense SW 6 SSW6
Interrupt SI07	Intrpt Mask, IM07	Check Carry Out CCP	Sense SW 7 SSW7
R00	R04	R08	R12
Interrupt SI10	Intrpt Mask, IM10	Bus Busy 0 BB0	Error Stop ERS
Interrupt SI11	Intrpt Mask, IM11	Bus Busy 1 BB1	Intrpt Pending NPT
Interrupt SI12	Intrpt Mask, IM12	Bus Busy 2 BB2	RW MM Test RWM
Interrupt SI13	Intrpt Mask, IM13	Bus Busy 3 BB3	Shift Done SHD
Interrupt SI14	Intrpt Mask, IM14	Bus Active 0 BA0	All Ones Sense OSI00
Interrupt SI15	Intrpt Mask, IM15	Bus Active 1 BA1	All Ones Sense OSI01
Interrupt SI16	Intrpt Mask, IM16	Bus Active 2 BA2	All Ones Sense OSI02
Interrupt SI17	Intrpt Mask, IM17	Bus Active 3 BA3	All Ones Sense OSI03
R01	R05	R09	R13
Interrupt SI20	Intrpt Mask, IM20	OE LB F/F OLB00	Zero Sense ZSI00
Interrupt SI21	Intrpt Mask, IM21	OE LB F/F OLB01	Zero Sense ZSI01
Interrupt SI22	Intrpt Mask, IM22	OE LB F/F OLB02	Zero Sense ZSI02
Interrupt SI23	Intrpt Mask, IM23	OE LB F/F OLB03	Zero Sense ZSI03
Interrupt SI24	Intrpt Mask, IM24	CE LB F/F CLB00	Zero Sense ZSI04
Interrupt SI25	Intrpt Mask, IM25	CE LB F/F CLB01	Zero Sense ZSI05
Interrupt SI26	Intrpt Mask, IM26	CE LB F/F CLB02	Zero Sense ZSI06
Interrupt SI27	Intrpt Mask, IM27	CE LB F/F CLB03	Zero Sense ZSI07
R02	R06	R10	R14
Interrupt SI30	Intrpt Mask, IM30	CE LB F/F CLB04	Zero Sense ZSI08
Interrupt SI31	Intrpt Mask, IM31	CE LB F/F CLB05	Zero Sense ZSI09
Interrupt SI32	Intrpt Mask, IM32	CE LB F/F CLB06	Zero Sense ZSI10
Interrupt SI33	Intrpt Mask, IM33	CE LB F/F CLB07	Zero Sense ZSI11
Interrupt SI34	Intrpt Mask, IM34	CE LB F/F CLB08	Three Sense TSI00
Interrupt SI35	Intrpt Mask, IM35	CE LB F/F CLB09	Three Sense TSI01
Interrupt SI36	Intrpt Mask, IM36	CE LB F/F CLB10	Four Sense FSI00
Interrupt SI37	Intrpt Mask, IM37	CE LB F/F CLB11	Four Sense FSI01
R03	R07	R11	R15

- The last group of 64 State flip-flops are not hardware but pseudo-flip-flops. These can be tested for either the true or complement condition but cannot be set and reset directly. A branch test on the first group of pseudo-flip-flops (Group 1, R08) causes a "hiccup" (one clock delay) if the test fails.

1.4.9 Action Request Servicing

There are 32 Action Request conditions in the MLP-900. The presence of an masked Target System Interrupt during a CEDE/WIN is also treated as an Action Request. Referring to Table 1-5, the first two Action Requests (AR's) provide a MINIFLOW entry for orderly start-up or shut-down when power is switched. The next five AR's (Priority 3-7) are set when high-priority errors or status conditions are detected in the Control Engine. Neglecting the Trace flip-flop, the next eight AR's function similarly for the Operating Engine. The four action requests labeled "Bad Addr" (Address) are set when an access is made to Main Memory and the bank addressed was not active in the system. This AR is also enabled if an I/O device is not present when addressed. The Language Board Action Request (Priority 21) can be used to indicate that a Main Memory address was out of the range previously defined as a boundary by optional memory protection (such as base or limit registers) which can be provided on OE Language Boards. The External Call Action Requests (Priority 23-26) enables units communicating with the External Buses to request servicing by the processor. The External Interrupt AR's provide a means for generating interrupts from sources which are not necessarily tied to the External Buses.

The order in which the Action Requests are listed in the table represents their priority; this is approximately the order in which servicing proceeds if more than one interrupt occurs. The first 16 AR's in the table are of such priority that a transfer out of sequence is initiated upon the occurrence of the Action Request. The External Bus Parity AR (Priority 16) is caused by external conditions but detected during internal transfers. The last 16 Action Requests in the table are initiated directly by conditions external to the processor. Breakout for this group is postponed until the first CEDE or TEXT ministep with a Wait mode is executed. Breakouts for all Action Requests force a transfer to a dedicated location in Control Memory, called the AR Entry location. Because of timing constraints, a one-clock-time delay (hiccup) occurs at all breakouts. Entry location assignments are given in Table 1-5. The AR Entry locations, with one exception, are placed in the first few locations of Read-Write Control Memory, two locations per entry. This provides complete flexibility in handling the request, since a BEAD (Branch/Extended Address) will normally be programmed as one of the two Ministeps to transfer into a "soft" AR service routine. The contents of the AR Entry locations must be initialized when Control Memory is loaded. One exception is the Power On Action Request, whose entrance location is in Read-Only Control Memory. A non-volatile MINIFLOW routine is used to bring the processor up from a cold start.

When an Action Request causes a breakout from a sequence, a return address is set into the Subroutine Return stack. For most detected error conditions, the current address is used, so that a retry may be attempted. In a few cases, the continuation address is saved. Except for Adder-Shifter Error and Extension Shifter Error, a return

Table 1-5. MLP-900 Action Request Characteristics

Name	Priority	Breakout	Special Action	AR Entry	Return	Mask	Lockout
Power On	1	Immed.	None	65280	No	No	ARL1
Power Off	2	Immed.	None	0	Current	No	ARL1
Parity, Odd CM	3	Immed.	Inhibit Clock	2	Current	No	ARL1,ARL2
Parity, Even CM	4	Immed.	Inhibit Clock	4	Current	No	ARL1,ARL2
Invalid CM ADDR	5	Immed.	None	6	Current	No	ARL1,ARL2
Stack Full	6	Immed.	None	8	Current	No	ARL1,ARL2
Stack Underflow	7	Immed.	None	10	Current	No	ARL1,ARL2
MINIFLOW Trace	8	Immed.	None	12	Continue	No	ARL1,ARL2,ARL3
Adder-Shifter Error	9	Immed.	None	14	Continue	ARM1	ARL1,ARL2,ARL3
Extsns-Shifter Error	10	Immed.	None	16	Continue	ARM1	ARL1,ARL2,ARL3
A Bus Parity	11	Immed.	Inhibit Clock	18	Current	ARM1	ARL1,ARL2,ARL3
B Bus Parity	12	Immed.	Inhibit Clock	20	Current	ARM1	ARL1,ARL2,ARL3
Extension Parity	13	Immed.	Inhibit Clock	22	Current	ARM1	ARL1,ARL2,ARL3
Mask Parity	14	Immed.	Inhibit Clock	24	Current	ARM1	ARL1,ARL2,ARL3
Exchange Bus Parity	15	Immed.	Inhibit Clock	26	Current	ARM1	ARL1,ARL2,ARL3
Extrnl Buss Parity	16	Immed.	Inhibit Clock	28	Current	ARM1	ARL1,ARL2,ARL3
Bad Addr Bus 0	17	Wait Mode	None	30	Current	No	ARL1,ARL2,ARL3,ARL4
Bad Addr Bus 1	18	Wait Mode	None	32	Current	No	ARL1,ARL2,ARL3,ARL4
Bad Addr Bus 2	19	Wait Mode	None	34	Current	No	ARL1,ARL2,ARL3,ARL4
Bad Addr Bus 3	20	Wait Mode	None	36	Current	No	ARL1,ARL2,ARL3,ARL4
Language Board	21	Wait Mode	None	38	Current	ARM2	ARL1,ARL2,ARL3,ARL4
MM Addr Compare	22	Wait Mode	None	40	Current	ARM3	ARL1,ARL2,ARL3,ARL4
Ext Call Bus 0	23	Wait Mode	None	42	Current	ARM4	ARL1,ARL2,ARL3,ARL4,ARL5
Ext Call Bus 1	24	Wait Mode	None	44	Current	ARM5	ARL1,ARL2,ARL3,ARL4,ARL5
Ext Call Bus 2	25	Wait Mode	None	46	Current	ARM6	ARL1,ARL2,ARL3,ARL4,ARL5
Ext Call Bus 3	26	Wait Mode	None	48	Current	ARM7	ARL1,ARL2,ARL3,ARL4,ARL5
Ext Interrupt 1	27	Wait Mode	None	50	Current	ARM8	ARL1,ARL2,ARL3,ARL4,ARL5
Ext Interrupt 2	28	Wait Mode	None	52	Current	ARM8	ARL1,ARL2,ARL3,ARL4,ARL5
Ext Interrupt 3	29	Wait Mode	None	54	Current	ARM8	ARL1,ARL2,ARL3,ARL4,ARL5
Ext Interrupt 4	30	Wait Mode	None	56	Current	ARM8	ARL1,ARL2,ARL3,ARL4,ARL5
Ext Interrupt 5	31	Wait Mode	None	58	Current	ARM8	ARL1,ARL2,ARL3,ARL4,ARL5
Ext Interrupt 6	32	Wait Mode	None	60	Current	ARM8	ARL1,ARL2,ARL3,ARL4,ARL5

to the normal sequence is still possible after the MINIFLOW service routine has been executed and the condition cleared.

Some of the Action Requests are inactivated by Mask bits in the MINIFLOW Status Word (described in paragraph 1.4.6). Two of the Mask bits, Action Request Mask (ARM) 1 and 8, inhibit two Action Request groups; others allow individual requests to be masked-off as shown in Table 1-5. Masked-off AR's are not capable of being set by hardware requests but can be set by MINIFLOW in the normal manner. Maskable Action Requests are active when the corresponding Mask bit is one (1), and inhibited for zero (0).

The Action Request system is mechanized with a multi-level priority structure which allows some AR's to interrupt certain others. A set of "Action Request Lockout" State flip-flops (ARL1-5) provide priority control. When a breakout occurs, the AR Lockout of that Action Request group is set. This inhibits other AR's in that group and lower priority groups. If an AR service routine is underway, and another Action Request occurs in a higher priority group, a breakout to the higher level is executed. This permits the execution of an interrupt while an interrupt is being serviced.

Locked-out AR's remain pending. Lower priority AR's break out as soon as all Lockout flip-flops of higher priority are reset.

When Control Memory parity errors are detected in two consecutive clock cycles, the master clock of the MLP-900 is turned off resulting in a machine halt. If the CM Partiy Action Requests are locked out, or if the TEST bit in the MINIFLOW Status Word is true, the halt is inhibited.

1.4.10 Control Engine Data Transfers

In Figure 1-4, the "Move Select" functional block represents the control and gating elements used to move information between the various portions of the Control Engine; obtain data from the Operating Engine Exchange Bus interface; and send data back to the Operating Engine over the Control Engine Data Bus. Transfers of data are, in general, accomplished by the Control Move ministep. Some subsidiary functions are performed by Move Select logic during the BRAD ministep, to execute the Copy A into B test mode for Branch type Control ministeps other than BRAD and BEAD and to perform the Conditional Copy operation during MAST. The MOVE ministep executes 16-bit and 8-bit transfers.

State flip-flops are addressed as a group of 8-bit or 16-bit registers by the MOVE ministep as well as one mode where a single State flip-flop can be used as the FROM address. The contents of any of the source addresses can be read out, but some destination addresses are not mechanized so that data may be read into them. This includes the State pseudo-flip-flops and Pointer Pseudo-registers 08 - 15. The Current Address Register contents cannot be altered by the MOVE ministep. Selection logic can pick up one of four 8-bit bytes from the Operating Engine Exchange Bus if the MOVE ministep specifies an 8-bit transfer. If the transfer is 16 bits, the most significant two bytes or the least significant two bytes will be picked up as a unit and transferred to the destination register. Control Engine Register addresses and a description of the MOVE ministep can be found in paragraph 2.3.8.

In addition the Control MOVE ministep, the Block Transfer (BLOT) ministep transfers data blocks of words to and from various groups of MLP-900 registers (in conjunction with the GENT ministep) or between the External Bus interfaces and register groups (paired with a CEDE or TEXT ministep). These functions are described in paragraph 2.3.6.

SECTION 2
MINISTEP FORMATS AND CONVENTIONS

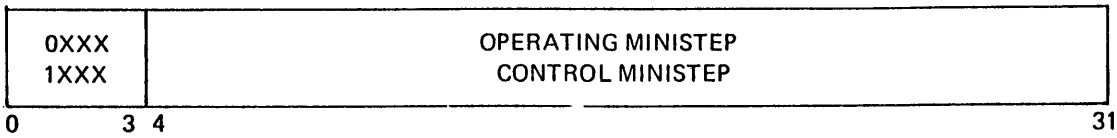


Figure 2-1. Basic Ministep Formats

2.1 GENERAL

MLP-900 ministeps are 32 bits long and are divided into Operating and Control varieties. One exception occurs where the 32-bit word immediately following a current Operating ministep is processed as data. This occurs when the B SELECT field in an Operating ministep specifies a Long Immediate Data word. A 36-bit operand is synthesized, using the LONG IMMEDIATE EXTEND field in the Operating ministep to supply the four high-order bits and the 32-bit successor word for the low-order part of the operand.

Bits are numbered 0 through 31, left to right. The leftmost bit in a word, field or sub-field is the most significant bit (MSB) for addressing, decoding, use as an operand, etc. The least significant bit (LSB) is to the right. The names of ministeps and fields are capitalized. Where the same bit is used to perform multiple functions in different modes of execution, the alternatives are shown in the drawing as sub-divided fields.

2.2 OPERATING INSTRUCTIONS

Operating ministep formats are shown in Figure 2-2. Abbreviations and symbols follow a common notation where possible. Notation conventions for describing Operating ministeps are shown below. Less commonly used designators are defined at their first appearance.

"A" is Operand A. "B" is Operand B. "$\overline{A}$" is the 1's complement of A. "COF1" is the "Initial Carry-Out" State flip-flop, which functions as a conditional carry-in for some arithmetic operations. "$\cdot$" denotes the logical product (AND) operation. "U" denotes the logical union (OR) operation. "E" denotes the Exclusive OR function. " $\leftarrow$ " (Left Arrow) denotes that the value of the right-hand term replaces the contents of the elements to the left.

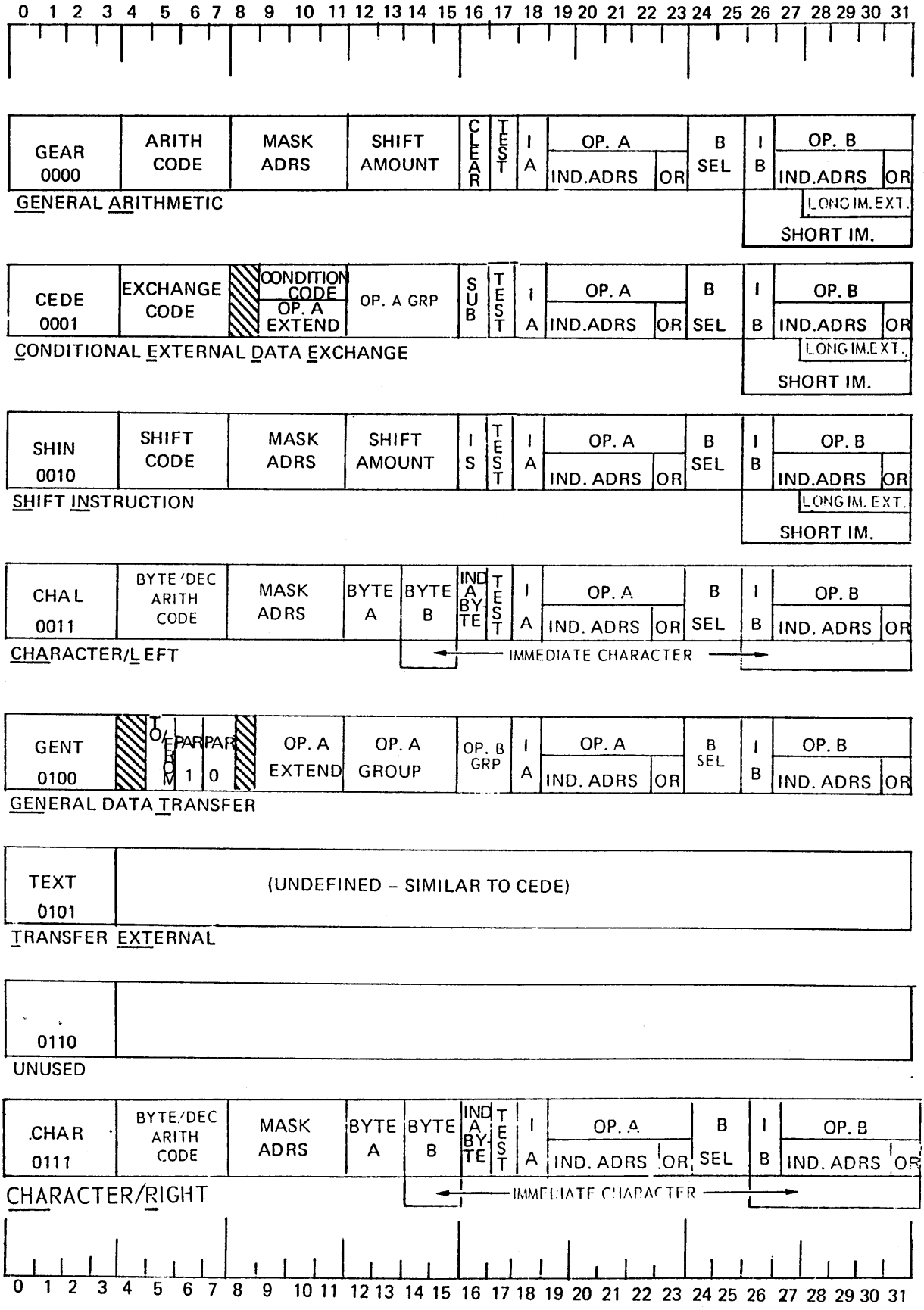


Figure 2-2. Operating Ministeps

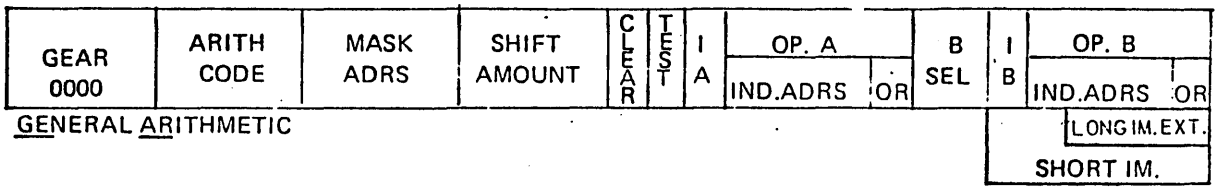


Figure 2-3. GEAR Format

2.2.1 GEAR - General Arithmetic

GEAR selects two operands and a Mask, routes them to the Primary Adder and specifies a shift of the result through the Primary Shifter. The input to Primary Shifter from the Primary Adder will have masked-out bits set to zero. The Zero Sense Pseudo (ZSP) State flip-flop reflects the zero state of this input. After the shift is performed, if the CLEAR bit is false (zero), then the masked-out positions are set to the original value found in the A operand. If the CLEAR bit is true (one), the output of the shifter is not affected by masking.

The OPERAND A and OPERAND B addressing structure of GEAR is repeated with some variations in the other Operating ministeps.

ARITHMETIC CODE (ARITH CODE), Bits 4-7: The Arithmetic Code field commands one of 16 binary arithmetic and logical operations on OPERAND A and B inputs. Table 2-1 lists the Primary Adder operations and coding.

Table 2-1. Arithmetic Codes and Functions

ARITHMETIC CODE	PRIMARY ADDER OPERATION	
09	$A \leftarrow A+B$	
14	$A \leftarrow A+\overline{B}+1$	(A-B; 2's complement subtract)
10	$A \leftarrow \overline{A}+B+1$	(B-A)
11	$A \leftarrow A+B+COF1$	(Conditional carry-in)
12	$A \leftarrow A+\overline{B}+COF1$	
13	$A \leftarrow \overline{A}+B+COF1$	
02	$A \leftarrow B$	(Clear and Add)
07	$A \leftarrow \overline{B}$	(1's complement)
05	$A \leftarrow A \cdot B$	(Logical AND)
03	$A \leftarrow A \cdot \overline{B}$	
01	$A \leftarrow \overline{A} \cdot B$	
06	$A \leftarrow AUB$	(Logical OR)
04	$A \leftarrow AU\overline{B}$	
00	$A \leftarrow \overline{A}UB$	
08	$A \leftarrow AEB$	(Exclusive OR)
15	$A \leftarrow AE\overline{B}$	(Compare)

MASK ADDRESS (MASK ADRS), Bits 8-11: Specifies one of 16 Mask Registers. Mask Bank selection is accomplished by the MBS State flip-flop. Masking is always enabled for GEAR, as follows:

- One (1) bits in Mask words are masked-in bits in the result
- Carry generation is suppressed in masked-out bits
- Carries propagate over masked-out bits

SHIFT AMOUNT, Bits 12-15: Specifies a single-length shift on the result word. Coding is given in paragraph 2.2 (SHIN ministep). Programmable shift amounts are left and right 0, 1, 2, 4, 6, 8, 12, and 16 bits. Bits shifted out of the result are lost, except that during left shifts, the last bit shifted out of the result word goes to the Shift Extension State flip-flop (SHE). Bits shifted in at the left end of the register are zeros or ones, as controlled by the Shift Out Sign State flip-flop (SOS). Bits shifted in at the right end of the word are zeros.

CLEAR, Bit 16: This bit is the mode specification for the masking operation. When CLEAR is true (1), masked-out bits (as translated by the SHIFT AMOUNT) are cleared to zero in the corresponding bits of the result (OPERAND A) register. When CLEAR is zero, masked-out bits are undisturbed and zeros are shifted into unmasked fields from masked-out bit positions. When CLEAR is false and shifts other than zero are specified, bits shifted into masked-out bit positions are lost. CLEAR true is a "No Op" when the TEST bit is true.

TEST, Bit 17: Inhibits transfer of the result word into the OPERAND A register when on (1). Permits testing status outputs of the result word without altering OPERAND A register contents.

INDIRECT A; B (IA; IB), Bits 18: 26: Indirect address control for OPERAND A and B selections. When the B SELECT coding specifies the General Registers, both IA and IB perform idential functions. IB is inactive for other B SELECT modes. When IA and IB are true (1), the respective indirect address (IND ADRS) field specifies one of the 16 Pointer Registers in the Control Engine. The contents of the five LSB's in the specified Pointer Register address General Registers as operands indirectly.

OPERAND A; B (OP A; OP B) Bits 19-23; 27-31: These address fields select one of 32 General Registers as OPERAND A and OPERAND B inputs. The OPERAND A mode is active when the IA bit is off (0). The OP B mode is active when the B SELECT mode specifies the General Registers for OP B and the IB bit is off.

INDIRECT ADDRESS A; B (IND ADRS), Bits 19-22; 27-30: Selects one of 16 CE Pointer Registers as an INDIRECT ADDRESS Register. When active, the least significant five bits of the specified Pointer address one of the General Registers. The A IND ADRS specifies a Pointer when the IA bit is one (1). The B IND ADRS field specifies a Pointer when B SEL specifies the General Registers (B SEL=0) and the IB bit is one (1); or when the contents of a Pointer are used as an 8-bit operand (B SEL=1).

OR (A; B), Bits 23, 31: Modifies the indirect address in the Pointer Register by logically OR'ing into the least significant bit (LSB) position. If the LSB in the Pointer Register is a one (1), the OR bit specification does not change the operand address. If LSB is zero (0), a one (1) OR bit will select the odd member of an even/odd register pair (normally used for double-register-length arithmetic).

B SELECT (B SEL), Bits 24 and 25: Selects one of four sources of Operand B inputs. The coding of the B SELECT Field is:

B SELECT Code	OPERAND B Input
0	General Registers
1	Pointer Registers
2	Short Immediate Data
3	Long Immediate Data

For General Register inputs (B SEL=0), the OP B specification is identical to that of the OP A field, both direct and indirect. B SEL=1 specifies the Pointer Registers, as addressed in the IND ADRS field, as an 8-bit auxiliary OP B input. This input is copied into the eight, LSB positions of the OP B input. OP B is zero in all other positions. B SEL=2 inputs bits 26-31 as a Short Immediate (literal) operand into the six LSB positions of OP B. Upper bits in OP B are zero. B SEL=3 specifies the Long Immediate Operand. The Long Immediate Operand is formed from the successor word from the Control Memory plus the 4-bit LONG IMMEDIATE EXTENSION (LONG IM EXT) to form up a complete, 36-bit, OP B input word. The 32-bit word from Control Memory goes into the 32 LSB positions of the OP B word and bits 28-31 in the OP B field are copied into the four MSB positions.

State Flip-Flops Affected. Several State flip-flops and pseudo-flip-flops may be active during GEAR. They are:

- Carry Out Pseudo (COP)
- Carry-out Flip-Flop 1 (COF1)
- Carry-out Flip-Flop 2 (COF2)
- Zero Sense Pseudo (ZSP)
- Zero Result Flip-Flop 1 (ZRF1)
- Zero Result Flip-Flop 2 (ZRF2)

(For SHIFT AMOUNTs other than zero)

- Shift Out Sign (SOS)
- Shift Out Flag (SOF)
- Shift Extension (SHE)

Their logic functions are shown in Table 2-2.

Table 2-2. State Flip-Flop Functions (GEAR)

FUNCTION	CONDITIONS	MECHANIZATION
Carry-out	ARITH CODES 09, 14, 10, 11, 12, 13	COF2 ← COF1 ← COP
Zero Result	All ARITH CODES Except Initial Carry-in (COF1)	ZRF2 ← ZRF1 ← ZSP
Zero Result With Carry-in	ARITH CODES 11, 12, 13	ZRF2 ← ZRF1 ← ZRF1 • ZSP
Sign Control	Right Shift ≠ 0	Bits shifted in bit 0 copy SOS
Significance Check	Left Shift ≠ 0	Bits shifted out of bit 0 are compared to SOS Any bit different from SOS sets SOF
Register Extension	Left Shift ≠ 0	Last bit shifted out of bit 0 goes to SHE

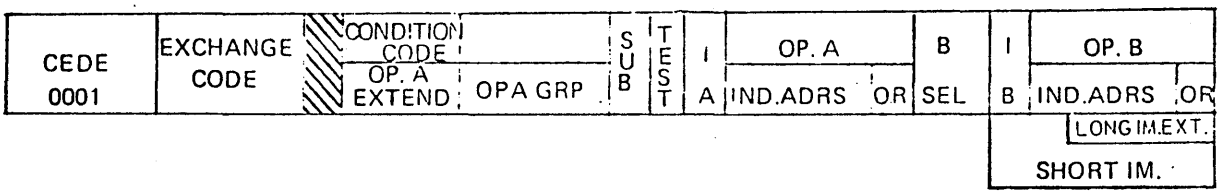


Figure 2-4. CEDE Format

2.2.2 CEDE-Conditional External Data Exchange

CEDE transfers data addresses and control codes between Main Memory and the Operating Engine External Registers. The active External Bus is specified in the MINIFLOW Status Word. CEDE types are mechanized to perform the sequences which fetch and translate target instructions, fetch and store operands and load MINIFLOW and processor initialization data. Corresponding timing and acknowledge inputs from Main Memory aid in sequence control. Some CEDE's place the MLP-900 Processor in the Wait mode. If the corresponding acknowledge signal has not been received from Main Memory, the Wait mode is entered unless an Action Request (AR) or, for the CEDE/WIN, a Target System interrupt is pending. Receipt of the correct acknowledge signal initiates execution of the CEDE ministep. The Wait mode is not entered if the acknowledge signal is present prior to the start of CEDE execution and no AR's (or Target System interrupts with CEDE/WIN) require servicing.

EXCHANGE CODE, Bits 4-7: Specifies one of the CEDE types for Execution. CEDE codes and functions are described in Table 2-3. There are three classes of CEDE ministeps:

- Memory Command Word Generation (CMD)
- Data Transmission (DATA)
- Combinatorial (COMB)

Table 2-3. CEDE Exchange Codes

CODE	TYPE	MNEMONIC	FUNCTION	DESCRIPTION
00	CMD	FIN	A, EO_{ADR} ← (A±B)	Fetch Instruction
01	COMB	WIN*	CM BRANCH ADR ← EI_{LB} A, EO ← (EI_{OAD}±B)** IR ← EI	Wait for Instruction; Execute MINIFLOW entry; Generate Address and Save; Fetch Operand; Load Target Instruction in Register
02	CMD	FOP	A, EO_{ADR} ← (A±B)	Fetch Operand
12	DATA	WOP	A ← EI	Wait for Operand and Load
08	COMB	WOF	A ← EI EO_{ADR} → B	Wait for Operand and Load; Fetch Operand
15	COMB	WON	A ← EI EO_{ADR} → B	Wait for Operand and Load; Fetch Instruction.
07	COMB	WIF	A, EO_{ADR} ← (EI_{OAD}±B)**	Wait for Indirect Word; Fetch Operand
03	CMD	SOP	EO_{ADR} ← (A±B)	Store Operand (Write Request)
09	DATA	WAS	EI, EO ← A	Wait for Addr. Acknowledge and Store Data
10	- - -	Unused		
11	- - -	Unused		
13	- - -	Unused		
14	CMD	ROW	EO_{ADR} ← EO_{ADR} (Generate Command Strobe)	Retry Output Word and update Retry Counter (CE Group 0-R15)
04	CMD	GAD	A ← (EI_{OAD}±B)	Generate Address and Save
05	CMD	RMW	EO_{ADR} ← (A±B)	Read/Modify/Write Cycle Request
06	DATA	WSS	Write Inhibit Output	Wait for Command Acknowledge and Suppress Store

*Inhibited if Action Request or Target System Interrupt is pending.
**Inhibited if operand fetch unnecessary.

Notes on Table Symbols and Callouts:

"EO" is an External Output Register.
"EI" is an External Input Register.
"ADR" is the SC-700 Memory Address field, bits 10-31 in the EO.
"CM BRANCH ADR" denotes a Control Memory MINIFLOW initialization entry (even location only).
"SE" is the Shift Extension Register.
"±" denotes an add or subtract operation, specified by the SUB field.
"OAD" is an operand address field extracted from an incoming word.
"LB" denotes the Operating Engine Language Boards.
"IR" denotes the OE Target Instruction Registers.

NOTE

All addresses destined for the External Output Register pass through (and may be modified by) the selected Language Board.

CONDITION CODE, Bits 9-11: The CEDE condition code field is usable by the MINI-FLOW programmer to provide specific inputs and control functions to Memory devices on the External Bus during command and combinatorial type CEDE ministeps. The condition code bits (9-11) are routed to bit positions 2-4, respectively, of the External Bus. These bits have no meaning to the MLP-900 Processor; however, they do have assigned functions for the SC-700 Memory unit, which are described in Appendix A.

TEST BIT, Bit 17: This bit inhibits transfer of data into the General Registers.

OPERAND A (Composite): Includes OP A EXTEND, OP A GROUP, IA and OP A fields as specified below. This addressing mode is active for WOP (Wait for Operand) and WAS (Wait for Acknowledge and Store).

OPERAND A EXTEND (OP A EXTEND), Bits 9-11: An extension of the OP A address field. Affords an 8-bit, OP A span of 256 directly addressable register locations. Used with the OP A GRP to access up to 1024 Auxiliary Register or OE Language Board locations.

OPERAND A GROUP (OP A GRP), Bits 12-15: Divides Operating Engine Registers (and the Control Engine interface) into related groups of registers for addressing purposes.

OPERAND A GROUP CODES

OP A GROUP	REGISTER ASSIGNMENT
0000	General Registers
0001	Data Mask Registers
0010	Miscellaneous
0011	Unassigned
0100	Aux. Bank 0
0101	Aux. Bank 1
0110	Aux. Bank 2
0111	Aux. Bank 3
10XX	Control Engine
1100	OE Language Board Bank 0
1101	OE Language Board Bank 1
1110	OE Language Board Bank 2
1111	OE Language Board Bank 3

The location assignment in the Miscellaneous Group (GROUP code 0010) is:

MISCELLANEOUS GROUP REGISTERS

LOCATION	NAME
00	Data Entry Switches
01	Main Memory Addr. Switches
02	Processor Addr. Switches
03	Unused
04	Primary Instruction Reg.
05	Secondary Instruction Reg.
06	Unused
07	Unused
08	External Register In 0
09	External Register In 1
10	External Register In 2
11	External Register In 3
12	External Register Out 0
13	External Register Out 1
14	External Register Out 2
15	External Register Out 3

SUBTRACT (SUB), Bit 16: Specifies performance of a two's complement subtraction. This bit is only active for CEDE varieties which combine two operands in the Primary Adder. When the bit is a one (1), two's complement subtraction is executed by the Primary Adder when the EXCHANGE CODE enables the operation. When the SUB bit is zero, an addition of the two operands is executed in the Primary Adder.

OPERAND A: B (Composite) Bits 18-31: Coding format is identical to GEAR ministep (refer to paragraph 2.2.1) except both operands are not specified for some CEDE varieties.

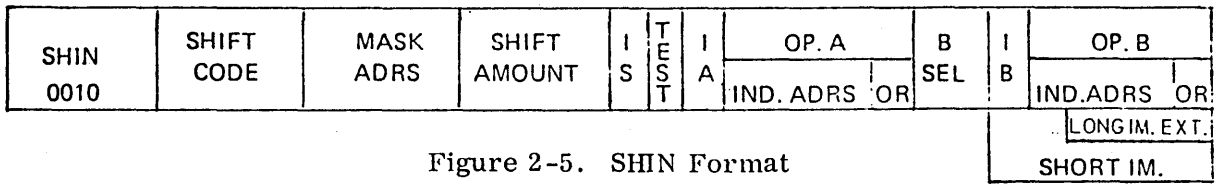


Figure 2-5. SHIN Format

2.2.3 SHIN-Shift Instruction

The SHIN ministep controls single and double register shifts in several modes and provides the basic functions for performing Multiply, Divide, and Normalize.

SHIFT CODE, Bits 4-7: Specifies one of 11 shift modes. A complete list of the SHIN codes and functions is given in Table 2-4. During double-length register pair shifts, the OP A field will normally address the even register and the odd address is implied. If the odd address is specified for a register pair shift, the OP A input and the OP B input are both the odd register, effectively operating as a one-register circular shift. When a SHIN ministep which calls for a double-length shift is executed, the Extension Shifter functions with the Odd General Registers. Single-length shifts may use any General Register as an operand. Some SHIN ministeps execute a double-length Shift with the Shift Extension Register. Both even or odd General Registers can be specified as OP A when the Shift Extension Register takes part in the Shift. OP B is used for addressing only with SHIN/MULTIPLY and DIVIDE, where it holds the Multiplicand and Divisor, respectively.

During NORMALIZE and DIVIDE, when the two-register pair specified by OPERAND A is used in a target language format which does not require a 72-bit word length, data in the even register (upper half) must be right justified and the data in the odd register (lower half) must be left justified. For multiply, the multiplier in the odd register is left justified. Multiplicands and Divisors in the OPERAND B register (or auxiliary input locations) must be right justified, as they are processed relative to the even register of the OPERAND A pair. A flow diagram of the MULTIPLY and DIVIDE operation is shown in Figure 2-6 and Figure 2-7. NORMALIZE shifts are sequenced by the CE Language Boards and are executed in a manner similar to indirectly controlled shifts. A tally of the shifted amount is generated in the Shift Control Pointer Register (P07) by outputs from the CE Language Board.

MASK ADDRESS (MASK ADRS), Bits 8-11: Specifies one of the 16 addressable Mask Registers as described in paragraph 2.2.1 (GEAR ministep). Masking affects only the input to the Primary Shifter and never the implied register of an even/odd register pair. Masked-out bits in the OP A register remain unchanged, since there is no CLEAR option on the SHIN ministep.

SHIFT AMOUNT, Bits 12-15: Specifies the number of bits to be shifted. A zero (0) in bit 12 specifies a right shift. A one (1) in this bit specifies a left shift. Coding of the rest of the field (bits 13-15) is treated as a 3-bit control number. Coding specifies the following shifts:

Shift Amount Code	0	1	2	3	4	5	6	7
Shift Span	0	1	2	4	6	8	12	16

INDIRECT SHIFT (IS), Bit 16: Controls the source of shift control inputs to the Shifters. If the IS bit is one (1), the five LSB positions in CE Pointer Register 07 specify the amount of the shift, except where one or more of the three high-order bits is a one (1). In this event, a 16-bit shift is executed.

Table 2-4. SHIN Ministep Shift Codes and Functions

Code	Shift Type	Direction	Mnemonic	Registers Shifted	Connections[1]	State F/F Operations
0	Even-Odd Reg. Long Linear	Right	SEOR	A, A(odd)	A←A(odd)	SOS←A(even)[2]
0	Even-Odd Reg. Long Linear	Left	SEOL	A, A(odd)	A(odd)←A	SHE←A, A(odd) ⊕ SOS←SOF[3,7]
1	Odd-Even Reg. Long Linear	Right	SOER	A, A(odd)	A(odd)←A	SOS←A(odd)
1	Odd-Even Reg. Long Linear	Left	SOEL	A, A(odd)	A←A(odd)	SHE←A, A ⊕ SOS←SOF
2	Single Register Linear	Right	SRR	A	None	SOS←A
2	Single Register Linear	Left	SRL	A	None	SHE←A, A ⊕ SOS←SOF
3	Dual Register Linear	Right	SDRR	A, A(odd)	None	SOS←A(even), SOS←A(odd)
3	Dual Register Linear	Left	SDRL	A, A(odd)	None	-
4	Even-Odd Long Circle	Right	REOR	A, A(odd)	A←A(odd)←A[5]	-
4	Even-Odd Long Circle	Left	REOL	A, A(odd)	A←A(odd)←A	-
5	Reg.-Extension Long Linear	Right	SRER	A, SE[6]	A←SE	SOS←A
5	Reg.-Extension Long Linear	Left	SREL	A, SE	SE←A	SHE←SE, SE ⊕ SOS←SOF
6	Extension-Reg. Long Linear	Right	SERR	A, SE	SE←A	SOS←SE
6	Extension-Reg. Long Linear	Left	SERL	A, SE	A←SE	SHE←A, A ⊕ SOS←SOF
7	Reg.-Extension Long Circle	Right	RRER	A, SE	A←SE←A	-
7	Reg.-Extension Long Circle	Left	RREL	A, SE	A←SE←A	-
8	Normalize	Left[4]	NORM	A, A(odd)	A←A(odd)	-
8	Normalize	Right	Undefined	-	-	-
9	Multiply[8]	Right[4]	MUL	A, A(odd)	A←A(odd)	SOS←A(even)
9	Multiply	Left	Undefined	-	-	-
10	Divide[8]	Left[4]	DIV	A, A(odd)	A←A(odd)	COP←COF1, A(odd)←COP
10	Divide	Right	Undefined	-	-	-
11-15	Unassigned	-	-	-	-	-

Notes:
1. Unless otherwise specified, bits shifted into a register from either end will be zero's.
2. All bits shifted into the register will be the same as the state of SOS if A is even, otherwise A(odd)←A(odd).
3. SOF is set if any bits shifted beyond bit 0 are different than the state of SOS. Once SOF is set, it can only be reset by MINIFLOW.
4. Shift direction is not implied in these ministeps, so it must be specified.
5. If A is even, a long circle shift occurs and if A is odd a short circle shift occurs.
6. SE is the Shift Extension Register.
7. ⊕ denotes "Exclusive Or".
8. A shift amount of 1 will normally be specified for multiply and divide.

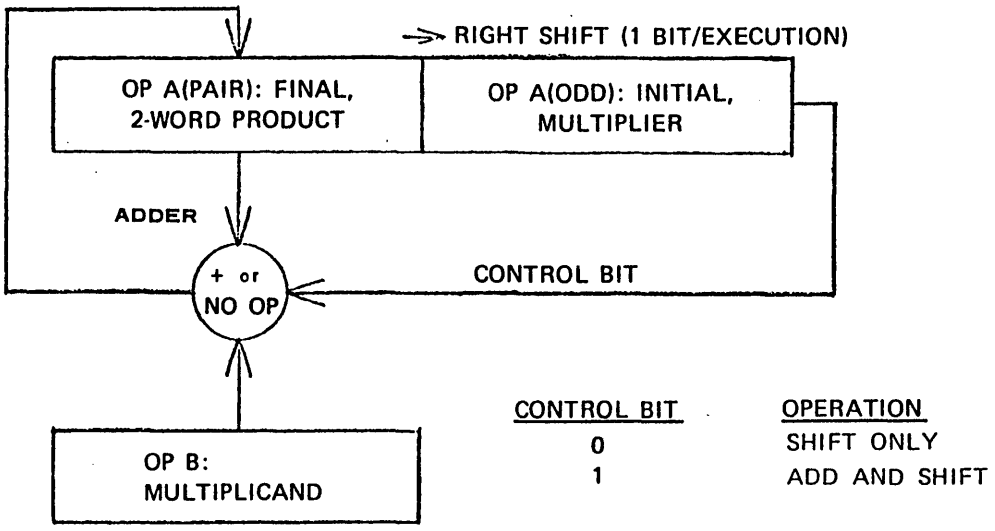


Figure 2-6. SHIN/MULTIPLY Flow Diagram

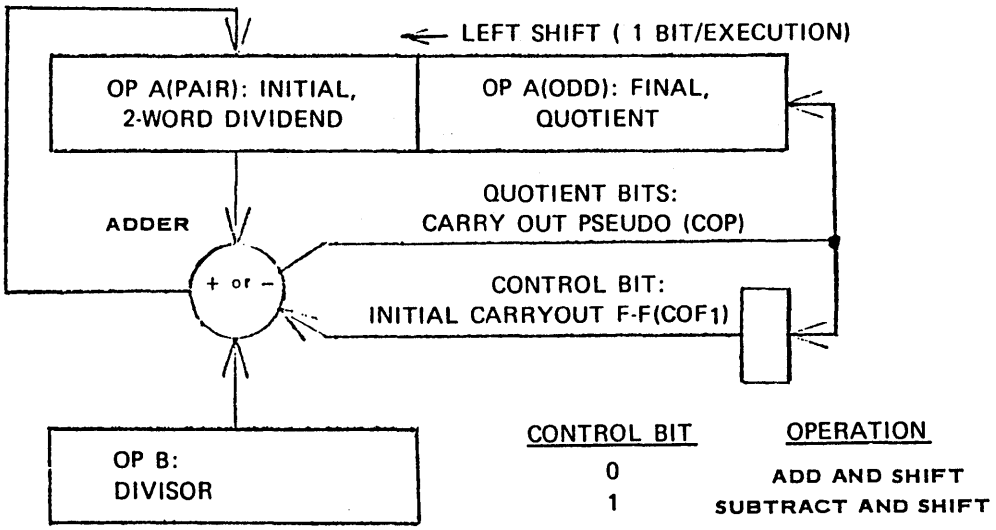


Figure 2-7. SHIN/DIVIDE Flow Diagram

A shift amount of 16 bits or less, and which is a power of two (1, 2, 4, 8, 16), can be executed in one clock cycle. Indirect Shift execution out of the Shift Control Register, for the five LSB's, is related directly to the highest-order bit that is on. Indirect Shifts are executed in the following order of precedence.

Bit Pattern (Bits 11-15)	Shift Amount
1XXXX	16
01XXX	8
001XX	4
0001X	2
00001	1

At the time of execution, the amount of the shift is subtracted from the contents of the Shift Control Register. By pairing a Branch ministep to test the "Shift Done" State pseudo-flip-flop, indirectly specified shifts up the counting capability of the Pointer Register can be iteratively executed. BRAD (Branch and Modify) Control ministeps should not be paired with indirect shifts for simultaneous execution.

TEST Bit, Bit 17: This bit inhibits transfer of data into the General Registers.

OPERAND A; B (Composite), Bits 18-31: Coding format is identical to paragraph 2.2.1 (GEAR Ministep) except both operands are not needed for some SHIN varieties.

State Flip-Flops Affected. The Shift Out Sign (SOS), Shift Out Flag (SOF) and Shift Extension (SHE) State flip-flops are active for most shifts. Logic mechanization is indicated in Table 2-4.

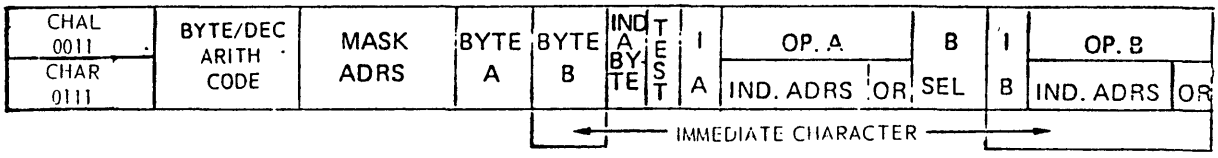


Figure 2-8. CHAL and CHAR Format

2.2.4 CHAL - Character/Left and CHAR - Character/Right

CHAL and CHAR operate on the 32 low-order bits of the Operand A and Operand B inputs (see Figure 2-8). These 32 bits are grouped into four 8-bit bytes (bits 4-11, 12-19, 20-27, and 28-35). OP A and OP B selection incorporates direct and indirect addressing to the byte level. For decimal operations, these ministeps treat the 8-bit byte as two BCD digits. Invalid signs and digits are detected and indicated by State flip-flops and pseudo-flip-flops (IDF, IDP, ISF, ISP). Carry correction is automatic.

BYTE/DECIMAL ARITHMETIC CODE (BYTE/DEC ARITH CODE), Bits 4-7: Selects decimal arithmetic, binary arithmetic and logical operations of 8-bit bytes. Table 2-5 lists the codes and functions of the Byte/Decimal Adder. Decimal arithmetic

operations treat operands as two 4-bit binary-coded decimal (BCD) characters. Logical and binary arithmetic operations are executed on 8-bit bytes. The OP A Byte normally holds the result of the operation (TEST bit off).

MASK ADDRESS (MASK ADRS), Bits 8-11: Specifies one of 16 Data Mask Registers for the Mask word. For the CHAL and CHAR ministeps, only the low-order eight bits of the Mask word are active. Masking operations and functions are identical to GEAR.

BYTE A: B: Bits 12 and 13; Bits 14 and 15: Specifies byte positions within the OP A and OP B inputs, respectively, for direct byte addressing.

INDIRECT A BYTE (IND A BYTE), Bit 16: Enables indirect specification of the Byte A address within the OP A word. When one (1), the two low-order bits of CE Pointer Register 00 provide the indirect Byte A Address. Pointer Registers 00 and 02 are decremented by one each execution of CHAL when byte A is indirectly addressed and TEST bit is zero. Pointer Register 00 is incremented and 02 is decremented by one each execution of CHAR under the same conditions. Branch ministeps can test Pointer Zero Sense and All Ones Sense, Three Sense and Four Sense pseudo-flip-flops and control the sequencing of operations to process character strings. For such operations, Pointer 00 is normally used to maintain the OP A byte position in the register word and Pointer 02 is used for the OP A character count in a variable-length string.

TEST, Bit 17: When true (1), no General Registers are changed and no Pointer Registers are modified by executing the ministep, but adder outputs can be tested by Control ministeps.

OPERAND A (Composite), Bits 18-23: Same format as GEAR.

B SELECT (B SEL), Bits 24 and 25: Selects one of four sources of Operand B inputs. Coding of the B SELECT Field is:

B Select Code	B Operand Input
0	General Registers with Direct B Byte
1	Pointer Registers
2	Immediate Character
3	General Registers with Indirect B Byte

For B SEL 0 and B SEL 1, the OP B inputs are the same as for the GEAR ministep. B SEL 2 specifies the 8-bit IMMEDIATE CHARACTER (IMMED CHAR) as the OP B input. The IMMED CHAR input is a composite of bits 14, 15, and 26-31 of the ministep in that order. B SEL 3 enables the two LSB's of Pointer Register 01 to address the byte position within the OP B input word. Pointer Register 03 is decremented and Pointer Register 01 is decremented (CHAL) or incremented (CHAR) by one whenever B SEL 3 is specified and TEST is false. Pointer 03 is normally used to hold the OP B byte count for processing a character string.

State Flip-Flops Affected

Several State flip-flops and pseudo-flip-flops, whose functions are shown in Table 2-6, are active during CHAL and CHAR. They are:

- Carry Out Pseudo (COP)
- Carry-out Flip-Flop (COF1)
- Carry-out Flip-Flop (COF2)
- Zero Sense Pseudo (ZSP)
- Zero Result Flip-Flop (ZRF1)
- Zero Result Flip-Flop (ZRF2)
- Invalid Digit Pseudo (IDP)
- Invalid Digit Flip-Flop (IDF)
- Invalid Sign Pseudo (ISP)
- Invalid Sign Flip-Flop (ISF)

Table 2-5. Byte/Decimal Arithmetic Codes

OP CODE		OPERATION
	DECIMAL	
00	A←A+$\underline{B}$	(Add)
04	A←A+$\widehat{B}$+1	(Sub)
01	A←A+$\underline{B}$+COF1	(Add, Conditional Carry)
05	A←A+$\widehat{B}$+COF1	(Subtract, Conditional Carry)
	BINARY	
08	A←A+$\underline{B}$	(Add)
12	A←$\underline{A}$+B+1	(Sub, 2's complement)
10	A←$\overline{A}$+B+1	(Sub, 2's complement)
09	A←A+$\underline{B}$+COF1	(Add, Conditional Carry)
13	A←$\underline{A}$+B+COF1	(Subtract, Conditional Carry)
11	A←$\overline{A}$+B+COF1	(Subtract, Conditional Carry)
	LOGICAL	
06	A←A•B	(AND)
14	A←AUB	(OR)
15	A←AEB	(Exclusive OR)
07	A←B	(Clear and Add)
02	Unused	
03	Unused	

"$\widehat{B}$" represents the 9's complement of B.

Table 2-6. State Flip-Flop Functions (CHAL and CHAR)

FUNCTIONS	CONDITIONS	MECHANIZATION
Carry-out	Byte/Decimal Arith. Codes 00, 04, 01, 05, 08, 12, 10, 09, 13, 11	COF2←COF1←COP
Register Zero	All Byte/Decimal Arith. Codes Except with Carry-in (COF1)	ZRF2←ZRF1←ZSP
Register Zero with Carry-in	Byte/Decimal Arith. Codes 01, 05, 09, 13, 11	ZRF2←ZRF1←XRF1 • ZSP
Invalid Digit	All Decimal Codes 00, 04, 01, 05	IDF←IDP If any of the four BCD operand digits is greater than 9, IDP is true
Invalid Sign	All Decimal Codes 00, 04, 01, 05	ISF←ISP If either of the LSD's of the two operands is less than 9, ISP is true.

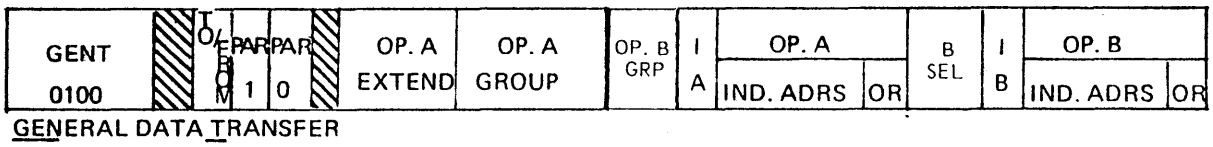


Figure 2-9. GENT Format

2.2.5 GENT - General Data Transfer

GENT performs direct transfers of the contents of Operating Engine Registers (see Figure 2-9). When the TO address specifies Control Engine and the GENT is immediately followed by a MOVE ministep (see p. 2-26) the pair can be used to transfer data to the Control Engine. If the FROM address specifies Control Engine while paired with the MOVE ministep data can be received from the Control Engine.

TO/FROM, Bit 5: Designates whether the extended OPERAND A field is a data source or a destination. When TO/FROM is one (1), OPERAND A is moved to the OPERAND B location. When TO/FROM is zero (0), the transfer reverses direction. Data addressed by the OP A GRP and OPA fields transfers to the B GRP and OP B address.

PARITY ONE (PAR 1), Bit 6: In conjunction with PARITY ZERO bit, controls parity logic when the MINIFLOW Status Word TEST MODE (MSW TM) bit is one. Used to perform tests on parity error detection logic. Coding of these two bits is shown below.

PARITY ZERO (PAR 0), Bit 7: Same function as PAR 1 bit, except zero parity is controlled.

PARITY ONE	PARITY ZERO	PARITY BITS STATE*
0	0	Normal Parity
0	1	0
1	0	1
1	1	1

*MSW TEST bit is on (1).

OPERAND A (Composite): Includes OP A EXTEND, OP A GROUP, IA and OP A fields as specified below.

OPERAND A EXTEND (OP A EXTEND), Bits 9-11: An extension of the OP A address field. Affords an 8-bit, OP A span of 256, directly addressable register locations. Used with the OP A GRP to access up to 1024 Auxiliary Register and OE Language Board locations.

OPERAND A GROUP (OP A GROUP), Bits 12-15: Divides Operating Engine Registers (and the Control Engine interface) into related groups of registers for addressing functions. The coding of the OP A GRP field is shown below:

OP A GROUP CODE	REGISTER ASSIGNMENT
0000	General Registers
0001	Data Mask Registers*
0010	Miscellaneous
0011	Unassigned
0100	Aux. Bank 0
0101	Aux. Bank 1
0110	Aux. Bank 2
0111	Aux. Bank 3
10XX	Control Engine
1100	OE Language Board Bank 0
1101	OE Language Board Bank 1
1110	OE Language Board Bank 2
1111	OE Language Board Bank 3

*Use of this code in a GENT for either a data source or destination will result in a no-op.

IA; OP A (Composite), Bits 18-23: Same as GEAR.

OPERAND B (Composite): Includes OP B GRP, IB and OP B fields as specified below:

OPERAND B GROUP (OP B GRP), Bits 16 and 17: Specifies register groups for OP B addressing. Several register-to-register transfers are not possible in the MLP-900, such as transfers from one Auxiliary Register to another Auxiliary Register address in the same clock cycle. Consequently, there are fewer OP B GRP assignments than OP A Group Assignments. They are:

OP B GROUP CODE	REGISTER ASSIGNMENT
00	General Registers
01	Mask Registers
10	Miscellaneous
11	Control Engine

Register addresses in the Miscellaneous (both OP A and OP B) Group are shown in Table 2-7.

Table 2-7. Miscellaneous Group Registers

LOCATION	NAME
00	Data Entry Switches
01	Main Memory Addr Switches
02	Processor Addr. Switches
03	Unused
04	Primary Instruction Reg.
05	Secondary Instruction Reg.
06	Unused
07	Unused
08	External Register In 0
09	External Register In 1
10	External Register In 2
11	External Register In 3
12	External Register Out 0
13	External Register Out 1
14	External Register Out 2
15	External Register Out 3

B SELECT (B SEL), Bits 24 and 25: Same coding as for GEAR.

2.2.6 TEXT-Transfer External

The TEXT ministep is not implemented for the initial version of the MLP-900, as system applications currently scheduled do not require direct communication to external devices other than SC-700 Main Memory Modules and a special purpose Data Exchange Unit (DEU) in the Memory Cabinet. When TEXT is implemented, retrofit to installed processors will be scheduled on a non-interference basis.

The proposed TEXT ministep format is similar to CEDE, with TRANSFER CODES tailored for communication with a broad range of devices and I/O interfaces.

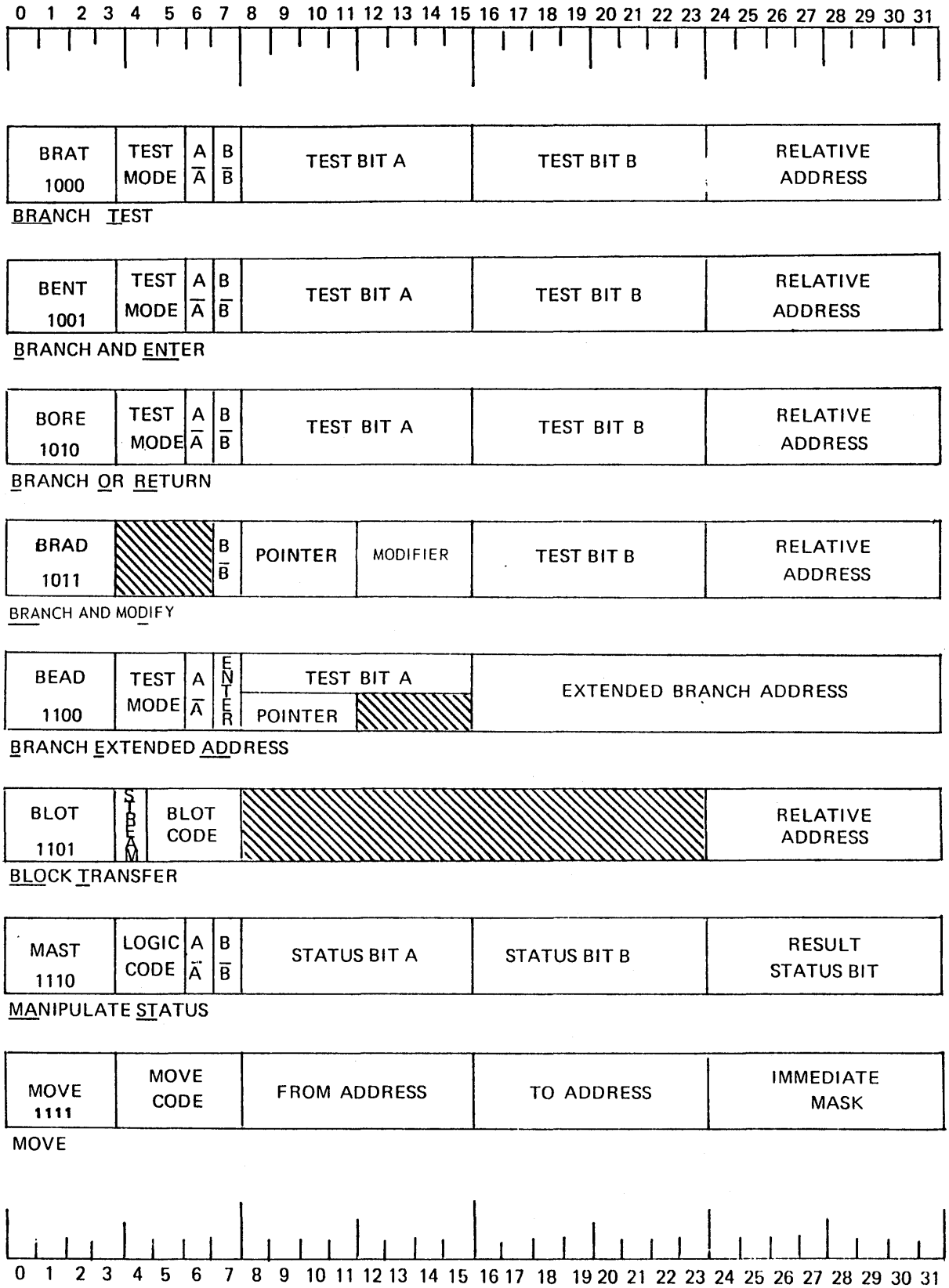


Figure 2-10. Control Ministeps

2.3 CONTROL MINISTEPS

Control ministep formats and field definitions are shown in Figure 2-10. In the tables in this section, A and B designators refer to TEST BIT A and TEST BIT B, respectively (instead of OPERAND A and OPERAND B). TEST BIT fields address the 256 State flip-flops and pseudo-flip-flops of the MLP-900 (refer to paragraph 1.4.1). Individual State flip-flops are addressed by subdividing the Test Bit, Status Bit and, in the case of MOVE/MOM, the From Address fields of Control ministeps according to the format shown in Figure 2-11 .

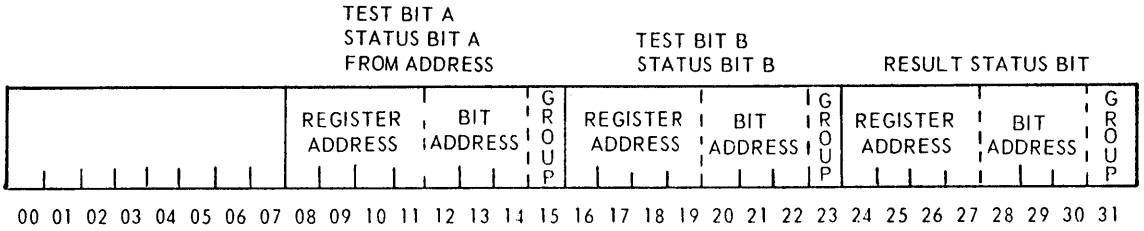


Figure 2-11. Control Ministep State Flip-Flop Address Format

All State Flip-Flops are synchronized with the MLP-900 internal clock (except Register 8 of Group 1) and do not change value during the execution of a ministep. They may be tested or moved freely. It is possible to get a "hiccup" (one cycle delay) when testing or moving any one or any combination of the eight flip-flops in Register 8 Group 1 (see Appendix C).

BRAT 1000	TEST MODE	A $\overline{A}$	B $\overline{B}$	TEST BIT A	TEST BIT B	RELATIVE ADDRESS

BRANCH TEST

Figure 2-12. BRAT Format

2.3.1 BRAT-Branch Test

BRAT is the least complex Branch ministep (see Figure 2-12. If the logic state specified by the TEST MODE, A/$\overline{A}$ and B/$\overline{B}$ fields is true, a branch address is generated by adding the continuation address and the contents of the RELATIVE ADDRESS field of the ministep. Otherwise, no branch occurs.

TEST MODE, Bits 4 and 5: Specifies the type of operation or logical combination of the TEST BIT A and B fields which are being sampled. When the TEST MODE condition is true (1), the branch is taken. "Move A to B" takes place whether branching occurs or not. The coding of the TEST MODE field is:

TEST MODE CODE	TEST CONDITION
0	TEST A and Move A to B (B←A)
1	TEST AUB (OR)
2	TEST A • B (AND)
3	TEST AEB (Exclusive OR)

A/$\overline{A}$; B/$\overline{B}$, Bit 6; Bit 7: Specifies the logical states of TEST BIT's A and B, for testing. A/$\overline{A}$, when one (1) specifies that the one side of the flip-flop is tested for the logical one (1) or set, condition. When A/$\overline{A}$ is zero (0), the test is for the zero (complement) output of the flip-flop true. The B/$\overline{B}$ field is coded in the same way. It performs the same operation for TEST BIT B, except for TEST MODE 0 (Move A to B). when it specifies whether a copy or complement move is to be taken.

TEST BIT A; B, Bits 8-15; Bits 16-23: Independently specify addresses of two of the 256 State flip-flops which are to be tested. As the two specifications are independent, the address of the same flip-flop can be coded in both fields.

RELATIVE ADDRESS, Bits 24-31: Specifies the amount by which the continuation address is to be altered. RELATIVE ADDRESS is added to the continuation address instead of the current address. Negative address amounts must be specified in 2's complement form. The span of the relative branch address is the continuation address +127, -128.

Figure 2-13. BENT Format

2.3.2 BENT-Branch and Enter

BENT is very similar to BRAT. The significant difference is that when the test is satisfied and the branch taken, a subroutine entry is executed. The continuation address is loaded into the Subroutine Return Stack. Paragraph 1.4.4 contains a description of the operation of the Subroutine Return Stack. Refer to Figure 2-13.

Figure 2-14. BORE Format

2.3.3 BORE-Branch or Return

BORE complements the BENT ministep. All other specifications are the same as BRAT. If the test is satisfied, the branch is taken. If the test fails, a subroutine return is executed by extracting a return address from the active Subroutine Return Register. The Subroutine Pointer Register is decremented by one and MINIFLOW execution proceeds from the return address point. Refer to Figure 2-14.

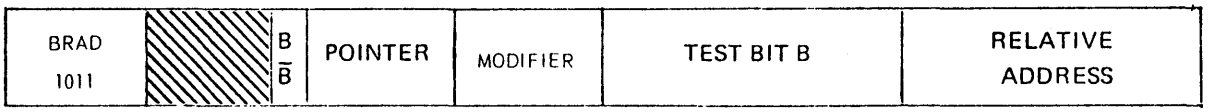


Figure 2-15. BRAD Format

2.3.4 BRAD-Branch and Modify Pointer

BRAD is a specialized branch ministep which is generally used for loop and count control. It operates in only one mode, and samples TEST BIT B for the true or complement state. If the test condition is satisfied, the branch is taken. Otherwise, no branch occurs. In either case, the pointer is modified. BRAD should not be executed in conjunction with a SHIN ministep when the INDIRECT SHIFT (IS) bit is on, or NORMALIZE code is specified. Refer to Figure 2-15.

POINTER ADDRESS (POINTER), BITS 8-11: Addresses the CE Pointer Register which is to be modified. One of the Counting Pointers (P00-07) must be addressed for BRAD to change the Pointer contents.

MODIFIER, Bits 12-15: Specifies the amount which is added to the Pointer Register when BRAD is executed. The MSB of the MODIFIER field is used as a sign, which gives a counting span of plus seven (+7) to minus eight (-8) in a single execution. Negative values must be specified in 2's complement form. Pointer modification is unconditional. If TEST BIT B is the Zero Sense flip-flop of the register specified by the POINTER field, BRAD can function as a loop or count control. Note that by using a MODIFIER value of other than plus or minus one, it is possible to pass through zero and end up with a non-zero residue in the Pointer. To test this situation, the Through Zero State pseudo-flip-flop is provided. Testing must take place during the same clock cycle that the modification occurs. The Through Zero output is developed so late in the execution cycle that a one-clock-cycle delay (hiccup) occurs when the branch is not taken.

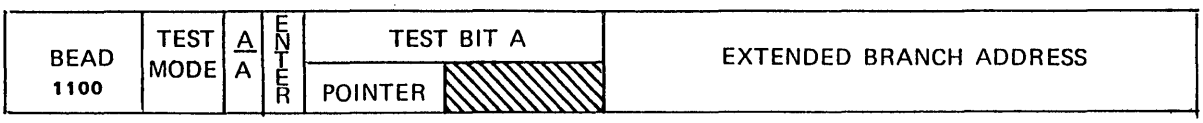


Figure 2-16. BEAD Format

2.3.5 BEAD-Branch-Extended Address

BEAD has four modes of operation. One of its major functions is to provide a capability for specifying branch addresses to all of Control Memory.

TEST MODE, Bits 4 and 5: Specifies test and addressing variations. TEST MODE functions are:

TEST MODE CODE	TEST FUNCTIONS
0	Conditional Absolute Branch
1	Absolute Branch plus Pointer
2	Continuation plus Pointer
3	Conditional Relative Branch

A/$\overline{A}$, Bit 6: This bit is active with TEST MODEs 0 and 3 and specifies the test condition for TEST BIT A as in the BRAT ministep.

ENTER, Bit 7: When true (1), and whenever a branch is taken, a subroutine entry is executed (the continuation address is loaded into the Subroutine Return Stack). TEST MODEs 1 and 2 are considered to always take a branch and therefore unconditionally cause a subroutine entry when the ENTER bit is true.

TEST BIT A, Bits 8-15: Active during TEST MODEs 0 and 3. Addresses one of the State flip-flops for testing, as in BRAT.

POINTER ADDRESS (POINTER), Bits 8-11: Active during TEST MODEs 1 and 2. Specifies the Pointer Register whose contents are to be used to develop the branch address.

EXTENDED BRANCH ADDRESS, Bits 16-31: Holds a 16-bit absolute or relative branch address to Control Memory. Active during TEST MODEs 0, 1, and 3.

The four variations are:

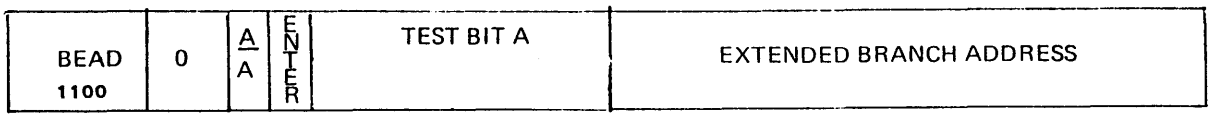


Figure 2-17. BEAD Format, Conditional Absolute Branch

2.3.5.1 Conditional Absolute Branch. If the test condition specified by the A/$\overline{A}$ field (bit 6) is true for the State flip-flop specified in TEST BIT A, the branch address in the Control Memory is specified absolutely in the 16-bit EXTENDED BRANCH ADDRESS field.

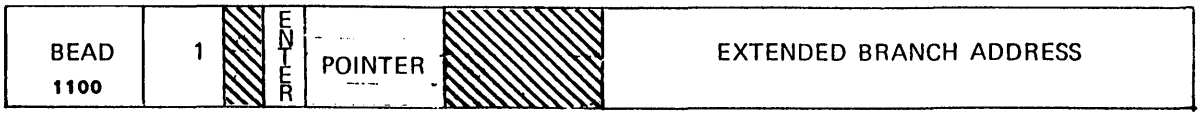


Figure 2-18. BEAD Format, Absolute Branch Plus Pointer

2.3.5.2 Absolute Branch Plus Pointer. When this BEAD is executed, the branch address is generated by adding the contents of the Pointer Register (POINTER) specified in bits 8-11 to the contents of the EXTENDED BRANCH ADDRESS field. The contents of the Pointer Register are treated as an 8-bit positive number.

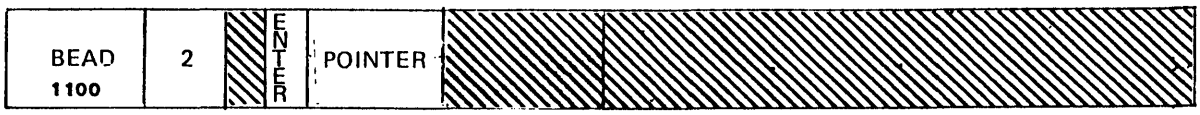


Figure 2-19. BEAD Format, Continuation Plus Pointer

2.3.5.3 Continuation Plus Pointer. In this mode of the BEAD Ministep, the absolute branch capability is not employed. The effect is similar to TEST MODE 1 in the preceding paragraph, except that the branch is taken by adding the contents of the specified POINTER Register to the MINIFLOW continuation address. The contents of the POINTER are treated as an 8-bit positive number.

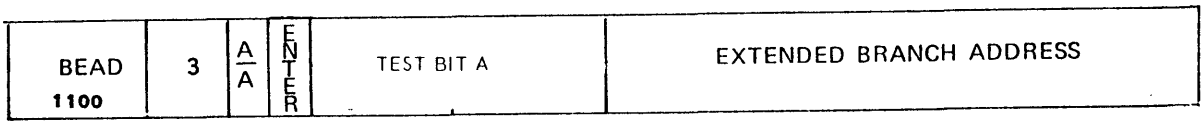


Figure 2-20. BEAD Format Conditional Relative Branch

2.3.5.4 Conditional Relative Branch. If the test condition specified is true, the Extended Branch Address is added to the continuation address to obtain the branch address. (Otherwise, the branch does not occur.) Wraparound may occur and is allowed. Thus the use of the proper 2's complement numbers in the Extended Branch Address field makes it possible to relatively address Control Memory in the negative direction.

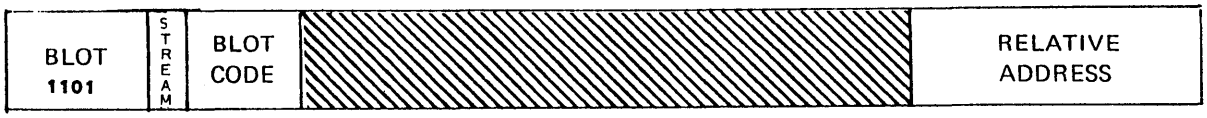


Figure 2-21. BLOT Format

2.3.6 BLOT-Block Transfer

BLOT is used to transfer multiple-word blocks of data within the processor, from an external source or to an external element. BLOT can also load in multiple (chained) blocks from external sources for processor start-up and initialization, Control Memory

overlay, etc. BLOT is usually teamed with CEDE, GENT, or TEXT ministeps, depending upon whether the move is to or from an external device or is an internal transfer. When transferring data to and from the Control Memory and the Subroutine Return Stack, the BLOT code controls the transfer directly, in conjunction with an Operating ministep (CEDE, GENT, OR TEXT). For Operating Engine transfers and when loading multiple blocks, BLOT functions as a sequence control for the operation. Refer to Figure 2-21.

STREAM CONTROL (STREAM), Bit 4: The STREAM bit is effective when a CEDE/WOP (Wait for Operand) is executed with a BLOT ministep. The function of the STREAM bit is to allow loading of multiple operand words from an input device which is itself designed for block transfers. (The block-transfer mode of read or write operation requires only a start address and an explicit or implicit block length specification.) Until Pointer 01 has a value of 1, STREAM prevents the Bus Busy State flip-flop from being reset on the execution of the CEDE/WOP, BLOT ministep pair so that another data word can be read in without sending a Command Word out.

BLOT CODE, Bits 5-7: Specifies the type of block transfer operation. Mnemonics for block transfers and their functions are:

CODE	MNEMONIC	FUNCTION
0	RCM	Read one block from CM; send to OE
1	WCM	Write one block into CM from OE with good parity
2	RSB	Read one block from Subroutine Stack; send to OE
3	WSB	Write one block into Subroutine Stack from OE
4	MOE	Move one block in OE
5	WBP	Write one block into CM from OE with bad parity
6	LMB	Load Multiple Blocks

RELATIVE ADDRESS, Bits 24-31: Specifies the increment for branching. These bits are added to the continuation address to form the branch address. The MSB of this field is used as a sign bit, in an identical manner as the Branch ministeps. Negative branch amounts must be in 2's complement form.

2.3.6.1 Pointer Operation. Pointer Register 00-03 are mechanized to function as counters and address generators during BLOT execution. Pointer 01 is used as a word counter. Pointers 02 and 03 are used together as a 16-bit counter to designate a Control or Operating Engine address. During single block transfers, Pointer 00 has no pre-assigned function and is available for use as an indirect Operating Engine address pointer. During multiple block load operations (LMB), Pointer 00 bit 3 is used as a chain designator, and bits 4-7 are used to hold a Register Group Code. When BLOT is executed, all three counters (P00, P01, P02/03) are decremented by one, except Pointer 00 is not decremented during BLOT/LMB.

2.3.6.2 Single Block Transfers. The point registers to be used must be initialized prior to the start of single block transfer operations. BLOT is designed to be used in a loop, with the Pointers determining when to exit from the loop. Each time BLOT is executed,

Pointer 01 (the word counter) is tested. When a count of one is present, the next ministep (or ministep pair) is taken from the continuation address. If Pointer 01 contains any count other than one, the next ministep (or ministep pair) is taken from the branch address. A word count of zero initially loaded into Pointer 01 may be used to transfer 256 words.

Single block transfers use a GENT; BLOT pair to transfer data between the Control Engine and the Operating Engine or between Operating Engine Register Groups. The GENT ministep can specify both source and destination register addresses directly, but indirect addressing is generally used, with P00 and P02/03 available for use as independent Pointers. Single block load operations move data into the processor from an external source, such as Main Memory or an I/O device. Execution of this function uses a CEDE or TEXT paired with BLOT ministep. Where the registers are in the Operating Engine, the CEDE and TEXT ministep OP A GRP and OP A EXTEND fields specify the data destination. When the destination registers and in the Control Memory, the appropriate BLOT code must be used to perform the transfer. Single block store operations are used to read a block of words out to an external device. If the Control Memory is the source of data, the CEDE/WAS is paired with the BLOT/RCM to perform the operation.

When the combined Pointer 02/03 is used to indirectly address registers in the Operating Engine and the Subroutine Stack in the Control Engine, care must be taken to prevent rollover of the address count within a register group. This is caused by attempting to load or read a register with a higher address than is contained in the group (for instance, addressing a General Register with an address greater than 31), or counting one of the Pointers used for addressing through zero and going around to count 255.

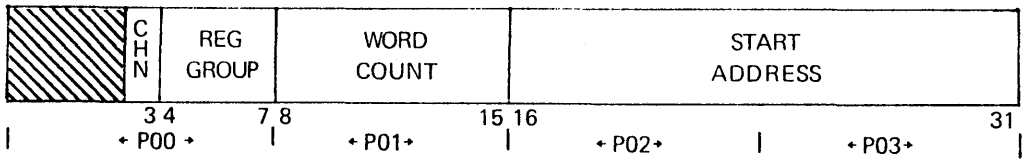


Figure 2-22. Load Control Word (LCW) Format (LMB Operations)

2.3.6.3 Multiple Block Load Operations. The Load Multiple Blocks (LMB) mode of BLOT utilizes the Load Control Word (LCW), as shown in Figure 2-22, as a header for blocks which are read in from an external source. BLOT/LMB, generally in conjunction with a CEDE/WOP, is used for initializing the processor from a cold start or for reinitializing the processor for a language changeover. The operation is basically a scatter/load technique. A sequence of self-defining data blocks can be loaded into all addressable register groups in the MLP-900 Operating Engine, Control Memory and the Subroutine Return Stack. The first word in any chained block must be the Load Control Word (LCW), which defines the number of words in the block, the register group they are to be loaded into and the starting address within the group. The LCW also includes a CHAIN (CHN) bit, which indicates whether or not there is another block following the current one.

To begin an LMB operation, Pointer 01 (the word counter) must be initialized to zero. When BLOT/LMB is executed, and Pointer 01 contains zero, the word on the OE Exchange Bus In is interpreted as a Load Control Word and is loaded into Pointers 00-03. Each time BLOT is executed and Pointer 01 is not zero, the data destination is determined by the address in Pointers 02/03, and the Register Group code in bits 4-7 of Pointer 00, as follows:

CODE	REGISTER GROUP
0000	General Registers
0001	Mask Registers
0010	Miscellaneous Registers (Operating Engine)
0011	Undefined
0100	Auxiliary Register Bank 0
0101	Auxiliary Register Bank 1
0110	Auxiliary Register Bank 2
0111	Auxiliary Register Bank 3
1000	Control Memory
1001	Subroutine Stack
1010	MINIFLOW Status Word
1011	Undefined
1100	OE Language Board Bank 0
1101	OE Language Board Bank 1
1110	OE Language Board Bank 2
1111	OE Language Board Bank 3

The branch address is used to access the next ministep pair each time BLOT/LMB is executed, except under the following conditions:

a. When the CHAIN bit is false (zero) and Pointer 01 contains a one, the continuation address is used.

b. When data is loaded into the Current Address Register, the new contents become the address of the next ministep pair.

The Current Address Register can be loaded during BLOT/LMB only when Pointer 00 contains a Register Group Code of 1010 (MINIFLOW Status Word) and the CHAIN bit is false (zero). If one attempts to load the MINIFLOW Status word when the CHAIN bit is true (one), data is blocked from entering the Current Address Register, and only the 16-bit MINIFLOW Status Register will be loaded. In this case, Pointer 01 would normally contain a count of one, which would be decremented to zero. The branch would be taken. The next BLOT/LMB executed would then interpret the next incoming data word as a new LCW.

Subroutine Stack Registers are addressed by using the sequential "SUBRTN REG" number, not the "R" number, shown in Table 2-8. Data destined for a Subroutine Stack Register is taken from the least significant 16 bits of the OE Exchange Bus In.

MAST 1110	LOGIC CODE	A / $\overline{A}$	B / $\overline{B}$	STATUS BIT A	STATUS BIT B	RESULT STATUS BIT

Figure 2-23. MAST Format

2.3.7 MAST-Manipulate Status

MAST sets or resets one State flip-flop based on a logical combination of the set or reset condition of two State flip-flops. The manipulation to be performed is specified by the LOGIC CODE field. Pseudo-flip-flops can be sensed but cannot be directly modified. Refer to Figure 2-23.

LOGIC CODE, Bits 4 and 5: Specifies the type of logical combination that is to be imposed on the RESULT STATUS BIT. The LOGIC CODE functions are shown below:

LOGIC CODE	OPERATION
0	If B test = 1, then RESULT←A (Conditional Move)
1	RESULT←AUB (OR)
2	RESULT←A • B (AND)
3	RESULT←AEB (Exclusive OR)

LOGIC CODE 0 provides a conditional move of the logic state of STATUS BIT A or its complement (A/$\overline{A}$ field) to the RESULT STATUS BIT if the STATUS BIT B test (as specified by the B/$\overline{B}$ bit) is true.

The other three codes provide the logical operations of OR, AND, and Exclusive OR.

A/$\overline{A}$; B/$\overline{B}$; Bit 6; Bit 7: These fields control sampling of the set or reset condition of STATUS BIT A and B, respectively. In conjunction with the LOGIC CODE field, these bits provide a capability to specify logical combinations of the two test bits, to derive set and reset inputs to the RESULT BIT State flip-flop.

STATUS BIT A; B, Bits 8-15; Bits 16-23: Specifies the inputs to the combining logic which drives the RESULT BIT. It is possible to code TEST BIT A and B for the same State flip-flop. Addressing format is the same as the TEST BIT A and B fields of the BRAT ministep.

RESULT STATUS BIT, Bits 24-31: Addresses the driven State flip-flop. The function specified in the LOGIC CODE field and in the A/$\overline{A}$ and B/$\overline{B}$ fields is used to reset or set the RESULT STATUS BIT, depending on the inputs and the logical mechanization of the RESULT flip-flop. Addressing format is the same as for the two STATUS BIT fields.

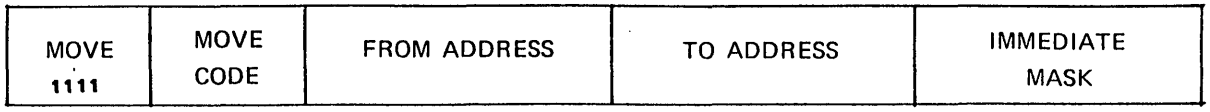


Figure 2-24. MOVE Format

2.3.8 MOVE-Control Engine MOVE

MOVE provides a means for moving data between registers in the Control Engine. MOVE can address all Control Engine Registers as if they are 8 bits wide. When a 16-bit transfer is specified, two 8-bit bytes are transferred in either normal or reversed address sequence. Register addressing format, except for the FROM address of the MOVE/MOM (Move One to Many) is shown in Figure 2-25. FROM in the MOVE/MOM is the same format for addressing individual State flip-flops shown in paragraph 2.3. Refer to Figure 2-24.

By addressing the OE Exchange Bus In (Registers 8-11, Group 3), data may be accepted from the Operating Engine when paired with (preceded by) a GENT or CEDE/WOP ministep. Conversely, by addressing the CE Data Bus Out (Registers 4-7, Group 3), data may be sent to the Operating Engine when paired with a GENT or a CEDE/WAS ministep.

MOVE CODE, Bits 4-7: Specifies one of six different modes of operation. MOVE CODE functions are:

MOVE CODE	OPERATION	MNEMONIC	MASKABLE	DATA LENGTH
000	Move ShortImmediate	MSI	Yes	8 bits
001	Move One to Many	MOM	Yes	8 bits
010	Move and Replace	MAR	Yes	8 bits
011	Move and Complement	MAC	Yes	8 bits
100	Move and Clear	MCL	Yes	8 bits
101	Move Double Byte	MDB	No	16 bits

For MSI, the FROM field is treated as an 8-bit data word and moved to the specified TO address. MOM uses FROM as a State flip-flop address (address format is the same as for the BRAT ministep). The state of the designated flip-flop is moved to all masked-in bits of the TO register. For MAR, FROM specifies an 8-bit register whose contents replace corresponding masked-in bits in the TO location. MAC is similar to MAR, but bits which are moved are complemented. Masked-out bits in the destination register are not changed. MCL differs from other MOVE's, as masked out bits are cleared to zero, otherwise it is identical to MAR. MDB allows two 8-bit bytes to be transferred. This MOVE is not maskable and is done on even/odd register pairs only. The two bytes will end up in normal or reversed sequence due to addressing mode. Register addresses in the TO and FROM Address fields are decoded as follows for 16-bit MOVES:

FROM ADDRESS	TO ADDRESS	SEQUENCE
Even	Even	Normal
Even	Odd	Reversed
Odd	Even	Reversed
Odd	Odd	Normal

FROM; TO ADDRESS, Bits 8-15; Bits 16-23: Specifies the source and destination registers in the Control Engine for MOVE. Register addresses in the Control Engine are divided into six groups, each containing up to sixteen 8-bit registers. Register address assignments are shown in Table 2-8. Addressing format is shown in Figure 2-25.

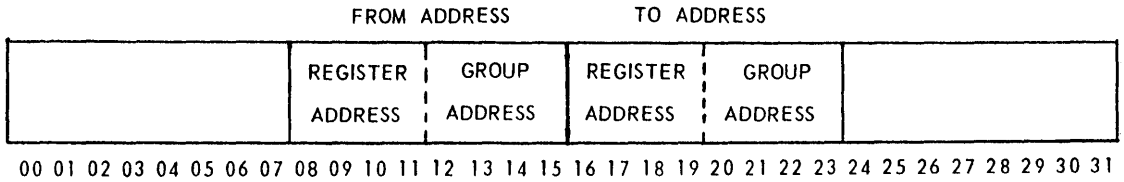


Figure 2-25. Register Addressing Format

The Subroutine Return Registers, although 16 bits wide, can also be addressed by 8-bit transfers. Access to the least significant eight bits of a Return address is obtained by specifying the corresponding odd register in the appropriate group (Groups 4 and 5) during an 8-bit MOVE. Byte transposition between upper and lower halves of the register can be accomplished by coding TO and FROM fields according to the rules given previously. When moving data to the Subroutine Return Registers, masking is not enabled. Because of the way in which the Subroutine Stack Registers are implemented, it is not possible to move the contents of one into another directly.

When Current Address Register is specified as a TO address, the MOVE ministep functions as a no-op.

IMMEDIATE MASK, Bits 24-31: Specifies which destination register bits will be affected by the MOVE. Masked-in bits allow transfer without alteration. For the MOVE/MCL (Move and Clear), masked-out bits are cleared to zero in the TO location. For other MOVE CODE options, masked-out bit positions remain in their original state. The Double-byte MOVE, MDB is not maskable.

Table 2-8. Control Engine Register Address Assignments

GROUP 0 - STATE FLIP-FLOPS

GENERAL INDICATOR F/F's			
R00	R01	R02	R03
LANGUAGE BOARD CONTROL F/F's		GEN. IND. F/F's	EXT. WRITE F/F's
R04	R05	R06	R07
ACTION REQUEST F/F's			
R08	R09	R10	R11
OE CONTROL F/F's		CE CONTROL F/F's	
R12	R13	R14	R15

GROUP 1 - STATE FLIP-FLOPS

TARGET SYSTEM INTERRUPT F/F's			
R00	R01	R02	R03
TARGET SYSTEM INTERRUPT MASK F/F's			
R04	R05	R06	R07
OE PSEUDO F/F's		OE & CE PSEUDO F/F's	
R08	R09	R10	R11
MANUAL & POINTER ALL ONES SENSE		POINTER ZERO SENSE	
R12	R13	R14	R15

GROUP 2 - POINTER REGISTERS

F/F POINTERS			
R00	R01	R02	R03
		SUBRTN CNTR	SHIFT CNTR
R04	R05	R06	R07
LB PSEUDO-POINTERS			
R08	R09	R10	R11
R12	R13	R14	R15

Table 2-8. Control Engine Register Address Assignments (continued)

GROUP 3 - MISCELLANEOUS

MINIFLOW STATUS		CURRENT ADDRESS REG	
R00	R01	R02	R03
CE DATA BUS OUT			
R04	R05	R06	R07
OE EXCHANGE BUS IN (Bits 4-35)			
R08	R09	R10	R11

GROUP 4 - SUBROUTINE STACK REGISTERS

R00 (SUBRTN REG 00)	R02 (SUBRTN REG 01)
R04 (SUBRTN REG 02)	R06 (SUBRTN REG 03)
R08 (SUBRTN REG 04)	R10 (SUBRTN REG 05)
R12 (SUBRTN REG 06)	R14 (SUBRTN REG 07)

GROUP 5 - SUBROUTINE STACK REGISTERS

R00 (SUBRTN REG 08)	R02 (SUBRTN REG 09)
R04 (SUBRTN REG 10)	R06 (SUBRTN REG 11)
R08 (SUBRTN REG 12)	R10 (SUBRTN REG 13)
R12 (SUBRTN REG 14)	R14 (SUBRTN REG 15)

APPENDIX A
SC-700 MAGNETIC CORE MEMORY

A.1 GENERAL

The SC-700 Memory cabinet holds up to eight 32,768-word 40-bit Memory Modules. Each of the four processor buses can communicate with up to 16 cabinets of eight Memory Modules. The nominal cycle time of the SC-700 unit is 700 nanoseconds. Read access time, after the receipt of an address and the initiation of a Read/Restore cycle, is a nominal 300 nanoseconds. Actual read access will not be this fast due to delays in the transmission of information to the processor External Bus Register. Assuming only one SC-700 Memory Cabinet, typical access times are on the order of 350 nanoseconds. There will be some minor variations due to differing transmission path lengths in the Memory Cabinet to the various Memory Modules. When more than one Memory Cabinet is used on a bus, the signals are "chained" through each cabinet by line receivers and drivers in the interface area. Additional circuit and transmission delays are incurred for each successive employment of this chaining technique. Two-way interlace of multiple Memory Modules is also available to improve effective access time in memory-limited applications.

The SC-700 Memory has four External Buses for communication with the outside world, and was designed to complement the MLP-900 Processor architecture. Any of the four MLP-900 External Buses may be interfaced to any of the four SC-700 Memory buses. It is even possible to connect more than one bus from a processor into a Memory Cabinet to different, or even the same Memory Unit. The SC-700 is designed to accept addresses from one bus, and directed by "Bus Offset" coding in the External Command Word, read data from or transfer data to another bus. The format of the External Command Word is shown in Figure A-1 below.

The SC-700 has two I/O interface mechanization modes. The Direct-Coupled interface (DCIO) is used where compatible signal levels, short cable runs and a benign electromagnetic environment allow. The Differential Interface Memory (DIM) Module uses differential current drivers through twisted pair wire to provide isolation and a capability to operate in high ambient noise environments,

Several card positions are reserved in the Memory cabinet for implementation of functions which are conveniently handled on the Memory Bus, but which are separated from the Memory functions. These facilities allow multiple use of the Memory Bus as an aid to system implementation. The address of a control element is distinguished from a Memory Location by the presence of the MMC signal (bit 4 of the External Command Word).

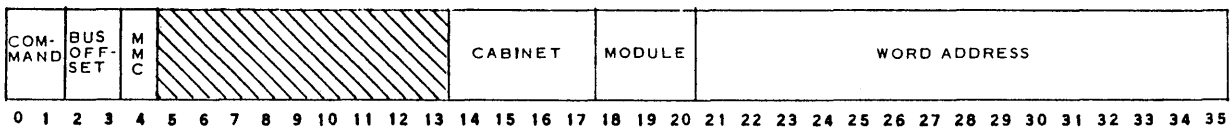


Figure A-1. External Command Word

A.2 EXTERNAL COMMAND WORD FORMAT

The COMMAND field, bits 00 and 01, are decoded as follows:

CODE	OPERATION
00	Unused
01	Write into Location
10	Read from Location
11	Read/Modify/Write to Location

After reading out data for a Read/Modify/Write cycle, the SC-700 stays on the bus until the write cycle is initiated by receipt of a data word. No other commands can be accepted until the sequence is completed.

The BUS OFFSET field, bits 02 and 03, contain the relative offset between the bus which received the External Command Word and the bus over which data is to be passed. The amount of the offset is added to the command bus number, modulo four, to obtain the offset bus address.

MAIN MEMORY CONTROL (MMC), bit 4, when true, indicates that the word address field in the External Command Word is not directed to one of the SC-700 Memory Units within the addressed cabinet. The cabinet has spare module slots for incorporating added facilities (such as protect key check logic or an operator's console interface) which are required to communicate over the same External Bus used for access to Main Memory.

NOTE

Bits 2-4 originate from the CONDITION CODE field of the CEDE ministep. Refer to paragraph 2.2.2.

Bits 5-13 are not assigned but are potentially available to perform subsidiary control functions (such as holding a memory protect key) where needed.

CABINET, bits 14-17, select one of up to 16 Main Memory Cabinets which can be chained on the bus.

MODULE, bits 18-20, select one of up to eight SC-700 Memory Modules which can be accommodated in each cabinet.

WORD ADDRESS, bits 21-35, defines the address of one of 32K word locations in the Memory Module (or special purpose control device in the Memory Cabinet).

In addition to the data bits of the External Command Word, there are several other lines that provide subsidiary logic and timing signals. The logic functions that are provided are:

- Parity - Four parity bits are transferred each time a data word is passed over the bus. Parity is generated at the data source and is maintained in the Memory. Parity checking is accomplished at the receiver. Each parity bit applies to a 9-bit byte. Parity is odd.

- Presence Acknowledge - This signal is generated by the addressed Memory Module. Lack of this signal notifies the originator of the External Command word that the addressed Memory Module is unavailable or inactive.

- Store Suppress - This signal is originated by the requestor of a Store or Read/Modify/Write cycle. It causes the Memory to restore the original data in the word and not continue in the storage mode. The usual cause for suppressing a store is the detection of a Memory Protect Address Violation at the MLP-900.

APPENDIX B
LANGUAGE BOARDS

B.1 GENERAL

There is no pre-defined logic on a Language Board (LB). The inputs can be logically combined in any desired manner and routed to any output. The LB's and the MINIFLOW for the realization of a Target language are closely related. The design of the LB's and the writing of a MINIFLOW must be coordinated to achieve the optimum in system performance.

The following paragraphs describe the nature and suggested uses of the Language Board input-output signals shown in Figure B-1. The numbers in parentheses following the signal name indicates the number of lines, or input-output pins associated with each class of signal. The arrows on the drawing indicate for each signal class whether it is an input, or an output.

B.2 SIGNAL DESCRIPTIONS

MASTER CLOCK (1) This is required for writing data into the registers associated with the address modification or memory protection function when implemented on the OE LB.

EXTERNAL IN REG. (36) These are the contents of the External In Register selected by the CEDE Bus bits in the MINIFLOW Status Word. When a WIN ministep is executed, the External In Register will usually contain an incoming target level instruction.

PRIMARY SUM (36) This the output of the OE primary adder. On a WIN or WIF, this will be the indexed target level address. On a GENT, these inputs may represent data which is to be loaded into registers associated with address modification or memory protection functions.

LB REG. ADDRESS (12) These provide a capability of addressing up to 1024 registers on the OE LB. Four of the inputs are in a 1 of 4 code representing one of four 256 word banks, and the other 8 are in normal binary. The former are decoded from the A group field in the CEDE and GENT ministeps, while the latter come directly from the 8 bits of the combined OP A and OP A Extend fields. State-of-the-art limitations currently limit the number of registers which can be implemented on one board to much less than the addressing capability.

LB REG. READ CONTROL (1) This signal causes the contents of the LB registers to be read out via the LB Data Bus. This signal is activated by a GENT ministep when the source address indicates a LB register. It may also be activated by a CEDE/WAS ministep.

LB REG. WRITE CONTROL (1) This signal causes data on the primary sum inputs to be written into the addressed LB register. This signal may only be activated by a GENT ministep, or a CEDE/WOP ministep.

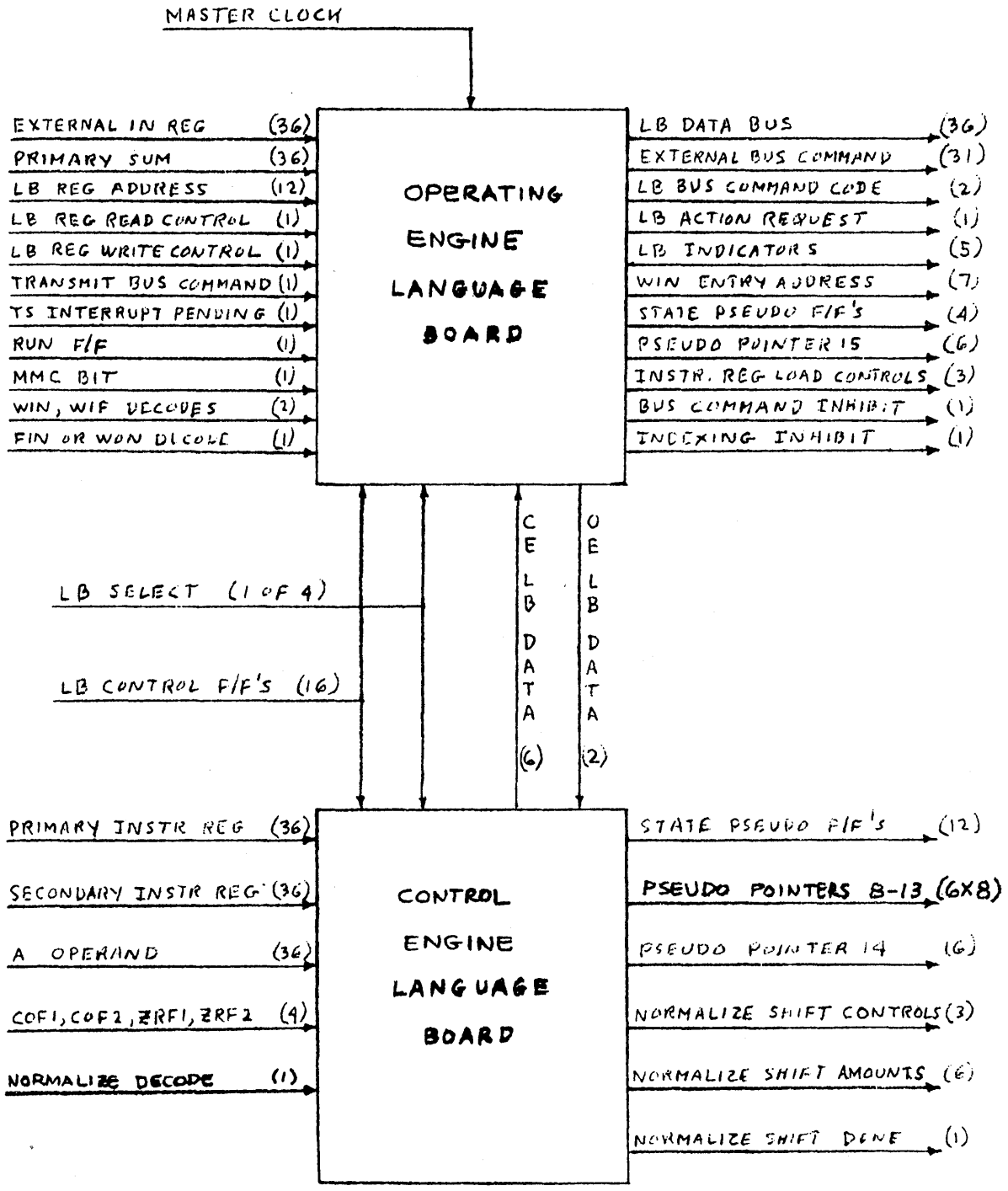


Figure B-1. Language Board Input/Output Signals

TRANSMIT BUS COMMAND (1) This signal is true whenever a Bus Command is to be set into an External Out register.

TS INTERRUPT PENDING (1) This signal indicates that a target system interrupt is pending.

RUN F/F (1) This input indicates that the target level computer is in the run state.

MMC BIT (1) This input indicates that a Bus Command is not intended for one of the SC-700 Memory units within the addressed cabinet (see Appendix A). It is required in order to disable any address modification and/or memory protection functions which may be implemented on the LB.

WIN, WIF DECODES (2) These two signals indicate that a CEDE/WIN or CEDE/WIF ministep is to be executed. They are used to enable generation of appropriate entry addresses and bus command codes at the proper time.

FIN or WON Decode (1) This signal indicates when an instruction fetch is to be made.

LB DATA BUS (36) On CEDE/WIN or CEDE/WIF ministeps, this word will contain the address field extracted from the target level instruction in the External In register. This word becomes the A operand input to the primary adder to enable target level indexing. On GENT minsteps, this word will represent the contents of the LB registers if these are addressed.

EXTERNAL BUS COMMAND (31) These signals are the least significant bits of the bus command which is to be loaded into the External Out register. The least significant 22 bits of this word are reserved for addresses, but the other bits may be used for other purposes such as protect keys, etc. The most significant five bits of the bus command word are generated elsewhere.

LB BUS COMMAND CODE (2) These two bits represent the type of memory cycle to be requested when a CEDE/WIN or CEDE/WIF ministep is executed. This will usually be derived from a partial decode of the instruction OP code. (See Appendix A - COMMAND Field of External Command Word.)

LB ACTION REQUEST (1) This action request may be used to indicate a limit violation, if a memory protection function is implemented on the OE LB, or other events as appropriate.

LB INDICATORS (5) These outputs represent 4 data signals and one control signal for setting the 4 LB Indicator State F/F's in R13 in group 0.

WIN Entry Address (7) These outputs specify an even entry address within the first 256 words of control memory. They are used only on a CEDE/WIN ministep.

NORMALIZE DECODE (1) This signal indicates that a SHIN/Normalize is being executed.

STATE PSEUDO F/F's (4) These outputs are the 4 OE LB F/F's, OLB00-OLB03, which are in R10 of the group 1 state F/F's.

PSEUDO POINTER 15 (6) These are the 6 LSB's of pointer 15. The upper 2 bits are omitted to save pins. This field will usually be used to indicate the index register to be used on the target level indexing operation performed by the CEDE/WIN or CEDE/WIF ministep.

INSTR. REG. LOAD CONTROLS (3) These 3 signals control the loading of the primary and secondary instruction registers on a CEDE/WIN ministep. One enables loading the primary instruction register from the External In register, a second enables loading the secondary instruction register from the External In register, and a third allows the primary instruction register to be loaded from the secondary instruction register.

BUS COMMAND INHIBIT (1) This signal provides a way to inhibit sending a bus command. This will usually be used on CEDE/WIN minsteps to prevent a memory cycle request from being issued for certain types of instructions.

INDEXING INHIBIT (1) This signal is used on a CEDE/WIN ministep to inhibit target level indexing for those classes of instructions which do not allow it. It is required since the CEDE/WIN would usually be written assuming that indexing will occur.

LB SELECT (1 of 4) This signal activates the outputs of one of the four pairs of Language Boards. Each pair of boards receives one select signal decoded from the two select bits in the MINIFLOW Status Word.

LB CONTROL F/F's (16) These signals go to both OE and CE Language Boards and provide a general way of controlling or modifying Language Board functions.

CE LB DATA (6) These inputs provide a generalized way to pass information available at the CE LB to the OE LB. One use would be to indicate the memory cycle type to be requested on a CEDE/WIF ministep.

OE LB DATA (2) These inputs provide a generalized way to pass information available at the OE LB to the CE LB.

PRIMARY INSTR. REG. (36) These inputs provide the source of data for most of the State Pseudo F/F's and Pseudo Pointers.

SECONDARY INSTR. REG. (36) These inputs provide an additional source of data for the State Pseudo F/F's and Pseudo Pointers. They will be useful for languages which have instruction words longer than 36 bits, or auxiliary instruction words of various types.

A OPERAND (36) This input is the A operand selected by the OE ministep. Its primary use will probably be for determining shift amounts for floating point normalization.

COF1, COF2, ZRF1, ZRF2 (4) These four signals are the carry out and zero sense flags. They can be used to generate condition codes or skip amounts for compare operations. These F/F's are located in R12 of group 0.

STATE PSEUDO F/F's (12) These outputs provide a way to sense selected bits or decoded states of the instruction word or other data available to the CE LB. These are CE LB F/F's CLB00-11 in R10 and R11 of group 1.

PSEUDO POINTERS 8-13 (6 x 8) These outputs provide a way to sense selected fields in the instruction words. These are pointers 8 through 13.

PSEUDO POINTER 14 (6) These 6 bits are the LSB's of pointer 14. It is shown separately to emphasize that it is shorter than the others.

NORMALIZE SHIFT CONTROLS (3) These three signals are applied to the prime shifters to control how many bits the operand is shifted when a SHIN/Normalize ministep is executed. The three signals are coded to allow shifts of 0, 1, 2, 4, 8, or 16 bits to occur.

NORMALIZE SHIFT AMOUNTS (6) These six signals are applied to the pointer adder, and are used to decrement the shift counter (pointer 7) by the amount of the shift.

NORMALIZE SHIFT DONE (1) This signal indicates that the current normalize operation has been completed. It will be used to determine when to terminate iterative execution of the SHIN/Normalize ministep.

APPENDIX C
STATE FLIP-FLOPS

C.1 GENERAL

State Flip-flops are a set of 256 addressable, single bit storage elements located in the Control Engine (CE), card slots 076, 077, 078 and 079. They are divided into two groups of 128 elements each. Each group is sub-divided into 16, 8-bit registers (Reg. 00 through 15) for addressing purposes.

All 128 elements of Group Zero are type "D" flip-flops. Half of Group One, the Target System Interrupts and Target System Interrupt Masks, are also type "D" flip-flops. The other half are called State Pseudo Flip-flops. They are not physical flip-flops, but true and complement data lines that are treated as outputs of flip-flops that may not be set or reset.

All State Flip-flops and Action Request Mask flip-flops (in the MINI-FLOW status Register) are reset (set to zero) in the power-up sequence and when MASTER CLEAR is depressed, except Pseudo Flip-flops which are asynchronous data lines. All State Flip-flops may be addressed as a source of data by Control Ministeps, and all State Flip-flops except State Pseudos may be utilized as destinations for data by Control Ministeps.

C.2 GROUP ZERO DESCRIPTION

Group Zero is made up of five basic sub-groups; General Indicators, Language Board Control, External Write, Action Requests and Control Flip-flops.

General Indicators; Registers 00-03 and Register 06. These elements are not dedicated and may be used for any general data storage or scratch pad functions.

Language Board Control; Registers 04 and 05. These elements are available for use as mode control inputs to the OE and CE Language Boards. When not used for Language Board Control they can function as General Indicators.

External Write; Register 07. These State Flip-flops are available as outputs to develop external signals independent of the External Bus Interface. When not assigned they can be used as General Indicators.

Action Requests; Register 08 through 11. Action Requests are MLP-900 Inner Computer interrupts. There are 32 Action Requests, referred to as AR00 - AR31. They are ordered on a priority basis, with AR00 being the highest priority. When two AR's request service simultaneously, the highest priority AR is taken. AR Masks and AR Lockout functions are used to control this priority scheme. There are eight AR Mask Flip-flops located in the MINIFLOW Status Register in bit positions 08-15 (ARM1-ARM8). When the mask bit is zero, its corresponding AR bit, or bits, are inhibited and will not respond to hardware set inputs. AR Flip-flops may be set or reset when directly addressed by Miniflow regardless of Masks. ARM1 masks AR08 through AR15, ARM2 masks AR20, ARM3 masks AR21, ARM4 masks AR22, ARM5 masks AR23, ARM6 masks AR24, ARM7 masks AR25, ARM8 masks AR26 through AR31. AR00 through AR07 and AR16 through AR19 are not maskable.

There are 5 AR lockouts that correspond to the five AR priority groups. These groups are AR00 and AR01, AR02 through AR06, AR07 through AR15, AR16 through AR21 and AR22 through AR31. Any Action Request, when taken, sets the lockout for its group. This lockout condition inhibits all ARs in the group and in all lower priority AR groups from being taken. Lockout differs from Mask in that lockouts do not inhibit the setting of the AR Flip-flop, with one exception (See Stack Underflow, AR06).

Action Requests (continued)

Power On, PON, (AR00) is set at the completion of the power on sequence. Entry is forced to the processor start-up routine in read-only memory. (Location 65280).

Power Off, POF (AR01) is set on the detection of loss of power. The address of the current Ministep is stored in the Subroutine Return Stack and MINIFLOW breakout to the "Power Off" AR routine is forced (Location 0).

Odd CM Parity Error, OPAR, (AR02) and Even CM Parity Error, EPAR, (AR03) are set on the detection of bad parity in the odd or even CM bank respectively. The address of the Ministep pair that initiated the error is stored in the Subroutine Return Stack and entry to the AR02 or AR03 service routine is immediately forced (Location 2 or 4).

Invalid CM Address, ICAD, (AR04) is set on the detection of an attempt to address a non-existent Control Memory location. The current address is stored in the Subroutine Stack and the AR04 service routine is entered (location 6).

Stack Full, STAF, (AR05) is set when Subroutine entry is requested and the Stack Pointer (P06) equals 14. The entry takes place and the normal continuation Ministep address is placed in Subroutine Stack location 15. The address of the first Ministep of the subroutine called by the branch is entered in Stack location 00 and the AR05 service routine is entered (Location 8).

Stack Underflow, STUF, (AR06) is set if a BORE Ministep requests a subroutine return when the Stack Pointer and ARL2 (lockout) equal zero. The BORE or the Ministep pair containing BORE is not executed and the Current Address is placed in Stack location 00. The AR07 service routine is entered (Location 10). NOTE: If ARL2 is set AR06 is inhibited from being set when a BORE is executed with the Stack Pointer equal to zero. This allows the return from the AR06 Service Routine.

MINIFLOW Trace, TRAC, (AR07) is set the first clock after the Initiate Trace Control FF (ITR) is set (if not in Wait status). The normal continuation address is stored in the Subroutine Return Stack and the AR07 service routine is entered (Location 12).

Primary ADDer/Shifter Error, PER, (AR08) is detected by comparing each bit on the Primary Bus with each bit on the Primary Check Bus. Any bad comparison sets the AR Flip-flop and transfers to location 14. The check takes place at the very end of the clock cycle and results

are not known in time to inhibit execution. If an error exists, the AR is set and the next Ministep pair is fetched but not executed. The address of this pair is stored in the Subroutine Return Stack and the AR08 service routine is entered. It is important to note that error results are clocked into the Result (Operand A) register, except when the TEST BIT of the OE Ministep is true.

Extension-Shifter Error, XER, (AR09) is detected by comparing the Extension Bus and the Extension Check Bus. Any bad comparison sets the AR Flip-flop and transfers to location 16. In all other respects AR09 processing is the same as AR08.

A Bus Parity, APAR, (AR10), B Bus Parity, BPAR, (AR11) Extension Bus Parity, XPAR, (AR12) Mask Parity, MPAR, (AR13) EXCHANGE BUS Parity, ECP, (AR14) and External Bus Parity, EBP, (AR15) indicate a sensed parity error in the associated data. When a parity error is detected, the execution of the Ministep or Ministep pair in process at the time of error is inhibited. The address of that Ministep or pair is stored in the Subroutine Return Stack and the proper AR routine is entered (Entry locations 18, 20, 22, 24, 26 and 28).

NOTE: Action Request 16 through 31 (AR16-AR31), when detected, are independently set but not acted on until a Ministep with a "Wait" mode is encountered. At that time Wait is inhibited, the current Ministep address is stored in the Subroutine Return Stack and the proper AR Routine is entered.

Bad Address Bus 0 through 3 (BAB0-BAB3) are inputs from the External Busses which indicate that the addressed device or storage area is either absent or disabled. This input is developed as the logical complement of the "Presence" input from active devices on each bus. When the Action request is taken, a Miniflow branch is made to CM location 30, 32, 34 and 36 for Bad Address 0 through 3, respectively.

Language Board Action Request, LBR, (AR20) occurs when a Language Board controlled event requires service. One use of such a service request could be upon limit violation detection in the OE Language Boards. Entry is forced to CM location 38 when LBR is detected.

Main Memory Address Compare, ADC, (AR21) is set when an equal compare is detected between the Main Memory Address switches on the Maintenance Panel and the address portion of an External Command Word in the External Output Register when a compare is requested by the proper settings of MM Read Addr, MM Write Addr, MM Data Addr and MM Inst Addr switches on the Maintenance Panel.

External Call Bus 0 through 3, CAB0-CAB3, (AR22-AR25) are interface lines that can be enabled by devices on the External Busses to request servicing.

External Interrupts, RUP0-through RUP5, (AR26-AR31) are available for use by an external device to generate interrupts separately from bus requests signals. One possible use might be a processor-to-processor communication interface.

Control Flip-Flops Registers 12 through 15 are indicators which monitor status and control important internal processor functions.

Carry Out, COF1, (CCFF00) copies carryout of the Decimal Arithmetic Unit and Primary Adder for GEAR and CHAR and CHAL arithmetic operations, and SHIFT MULTIPLY and DIVIDE ministeps. It effectively is set and reset by the Carryout Pseudo (COP) signal.

GEAR types for which COF1 (and COF2) are enabled are: 09, 10, 11, 12, 13 and 14 (Arithmetic codes).
CHAD codes for which COF1 (and COF2) are enabled are: 00, 01, 04, 05, 08, 09, 10, 11, 12 and 13 (Arithmetic codes).

Carry Out 2, COF2, (CCFF01) is used to store the previous contents of COF1 whenever it is changed by hardware.

Zero Result 1, ZRF1, (CCFF02) is driven by the logic which develops the Zero Sense Pseudo (ZSP) signal for all varieties of GEAR and CHAL and CHAR. Except as noted below it is set only on the detection of a Zero Result output (masked-in portion) of the Primary Adder.

When ZRF1 logic is modified for GEAR ARITH CODES 11, 12 and 13 and CHAL,CHAR ARITH codes 01, 05, 09, 11 and 13 (Conditional Carry In) the condition of ZRF1 is the logical product (AND) of its current state and the Zero Sense Pseudo (ZSP) output. That is:

$$ZRF1 = ZRF1 \cdot ZSP$$

This assists in multiple length operations.

Zero Result 2 ZRF2, (CCFF03) copies ZRF1 during those Ministep varieties when ZRF1 is enabled for set and reset inputs.

Invalid Digit, IDF, (CCFF04) is set during CHAL and CHAR when the most significant 4 bits of the A or B Byte or the least significant 4 bits of the A or B Byte (after being masked) are greater than nine (9).

Invalid Sign, ISF, (CCFF05) is set during CHAL and CHAR if the least significant 4 bits of A or B Byte (after being masked) are 9 or less.

Shift Out Sign, SOS, (CCFF06) is set by program. On any non-circular right shift in the primary shifter the state of SOS indicates the value of the shifted in bits. See Shift Out Flag explanation for an additional function of SOS. SOS allows convenient shifting of one's or two's complement numbers. It is reset to zero by CEDE WIN.

Shift Out Flag, SOF, (CCFF07) is set on non-circular left shifts if any bit shifted out of the OPA register does not equal the state of SOS. This Control Flip-flop is reset to zero by the CEDE WIN Ministep.

Last Bus In, LBI, (CCFF08) when TEXT is implemented, will be automatically set on TEXT and reset by CEDE Ministeps. As TEXT is not currently defined, it will always be zero unless set by program.

Shift Extension, SHE, (CCFF09) will hold the last bit shifted out on any non-circular left shift. It is reset to zero on CEDE WIN.

Mask Bank Select, MBS, (CCFF10) is not changed by hardware. If 0, it specifies Mask Bank 0 is active. Mask Bank 1 is active if set to 1.

Memory Bus Inhibit, MBI, (CCFF11) when set, causes the memory to ignore bus commands. Bus commands issued after MBI is set are not executed but remain pending. Because the memory does not respond to the command, a hang up will occur on any subsequent Wait type CEDE. If more than one bus command is issued while MBI is on, the one issued last will be the one executed after MBI is reset.

MBI can be set or reset by any of the CE Ministeps which affect State F/F's, and also will be set when the STEP switch on the Maintenance Panel is set to ON and the STEP TYPE switch is in the MEM position. MBI resets in the stepping mode when the Program Start pushbutton is depressed, but automatically sets one clock later.

Language Board Indicators, LI0, LI1, LI2, and LI3, (CCFF12, CCFF13, CCFF14, and CCFF15) are set by the OE Language Boards.

Single Instruction Interrupt Inhibit, SII, (CCFF16) is set by Miniflow. It inhibits the recognition of Target System Interrupts for one target level instruction. It is reset by the CEDE WIN Ministep.

Target System Interrupt Lockout, SIL, (CCFF17) is set by Miniflow. It inhibits recognition of all Target System Interrupts until reset by Miniflow or the Master Clear Switch.

Control Flip-Flops (continued)

Memory Command 0, MC0, (CCFF18) and Memory Command 1, MC1, (CCFF19) are set by hardware to indicate what the last CEDE memory command was.

MC 0	MC 1	
0	0	no-op
0	1	clear write
1	0	Read restore
1	1	Read Modify write

Clock Control, CKC, (CCFF20) when set indicates system clock is running and when reset indicates clock is off. It is set in the power up sequence and when Miniflow Start, Miniflow Start from Switches, or Program Start is depressed. It is reset to zero when Miniflow Stop or Master Clear is depressed and on certain hardware errors. (All switches are located on Maintenance Panel).

Run Flip-flop, RUN, (CCFF21) when reset inhibits all bus commands that apply to Target Level Instruction fetch operations, i.e., CEDE/FIN and the fetch half of CEDE/WON. It is set by Program Start and reset by Program Stop Switches.

Check Test, CKT, (CCFF22) when set causes the LSB of the shift code to the primary shifters to be forced to a 1. This can be used in diagnostics to cause a non-comare of the primary and check adder shifters outputs and thus cause a PER or XER action request. CKT is set by MINIFLOW and will automatically reset after one clock.

Initiate Trace, ITR, (CCFF23) is set by Miniflow. It causes the TRAC AR (AR07) to be set on the clock of any non-WAIT Ministep. ITR resets on the next clock after TRAC AR (AR07) sets.

Action Request Lockouts, ARL0, (CCFF24); ARL1, (CCFF25); ARL2, (CCFF26); ARL3, (CCFF27) and ARL4, (CCFF28) are set when AR entry is made in order to inhibit the recognition of any AR in that priority group or any lower priority group.

CCFF29 is not used.

Retry Counter Zero, TRC0, (CCFF30) and Retry Counter One, RTC1, (CCFF31) are used as a two bit binary counter which is incremented by one on each RETRY CEDE. Wrap around occurs at '11'.

C.3 GROUP ONE DESCRIPTION

Target System Interrupts, registers 00-03 (SI00-31)

Target System Interrupt Masks, registers 04-07 (IM00-31)

These flip-flops are set and reset by Miniflow program only. When corresponding Interrupt and Interrupt Mask bits are set (i.e. SI00=1, IM00=1) a Target System Interrupt is detected. A Miniflow branch is made to CM location 62 when a CEDE WIN is encountered, Single Instruction Interrupt Inhibit (SII, CCFF16) is zero, and Target System Interrupt Lockout (SIC, CCFF17) is zero.

State Pseudo Flip-flop Registers 08-15

Register 08; grouped in this register are all pseudo data inputs that occur so late in the clock cycle that a Move (all CE Ministeps that move data) or Test and Branch cannot be accomplished before the clock cycle completes. To overcome this whenever Register 08 is Addressed, a one clock delay (hiccup) occurs to allow the data time to settle (on a Test and Branch a hiccup only occurs when the test fails). Carry Out (COP), Zero Sense (ZSP), Invalid Digit (IDP), and Invalid Sign (ISP) have corresponding Control Flip-flops which are set by the pseudo inputs (COP-COF1, ZSP-ZRF1, IDP-IDF, and ISP-ISF).

Through ZERO (THZ) indicates that the Pointer Register currently addressed by a BRAD Ministep has overflowed or underflowed.

Wait AR Pending (WAR), indicates that one or more of the Action Requests AR16-AR31 is set, and is not affected by the Action Request mask or lockout settings.

Normalize Shift Done (NORD), is generated by the CE Language Board to indicate that the current Normalize Shift is complete.

Check Adder Carry Out (CCP), is the carry out sense line of the Check Adder.

Register 09; this register indicates the status of the four External Busses, Bus Busy (BB0-BB3) and Bus Active (BA0-BA3).

Bus Busy and Bus Active are set at the beginning CEDE of any input or output CEDE Pair (FIN-WIN, FOP-WOP, SOP-WAS, etc.).

For output CEDE Pair types Bus Busy is reset when the output device responds with Address Acknowledge. Bus Active is reset when the second CEDE of the output CEDE PAIR is executed. For input CEDE

State Pseudo Flip-Flops (continued)

PAIR types Bus Busy is reset when the input device has completed transfer of the input Data Word to the External Register and indicates Data Available to the OE. Bus Active is reset when the second CEDE of the input CEDE PAIR is executed.

Register 10, (Bits 0-3) comprise the OE Language Board Flip-flops (OLB00-OLB03). These inputs are OE Language Board status inputs to the CE.

Register 10, bits 4-7 and Register 11 comprise the CE Language Board Flip-flops. These inputs are CE Language Board Status inputs to the CE.

Register 12, Sense Switches (SSW0-SSW7); is made up of manual sense switch inputs from the Maintenance Panel.

Register 13, Bit 0 is the Main Memory Error Stop Switch (ERS) input from the Maintenance Panel.

Bit 1 is the Target System Interrupt pending (NPI) indicator. (Masked off interrupts are not considered pending.)

Bit 2 is Main Memory Read Write Test Switch (RWM) from the Maintenance Panel.

Bit 3, SHift Done (SHD) indicates that the current INDIRECT SHIFT is complete.

Bits 4-7, (OSI00-OSI03) indicates that the respective Pointer Register (00-03) equals all ones.

Register 14, Bits 0-7, and Register 15, Bits 0-3 (ZSI00-ZSI11) indicate that the respective Pointer Register (00-11) equals Zero.

Register 15, Bits 4,5 indicate that Pointer Register 00 and 01, respectively, contain the value three.

Register 15, Bits 6,7 indicate that Pointer Register 00 and 01, respectively, contain the value four.

NOTE: For sensing Pointer equal to zero, all ones, three, or four, it should be noted that the Pointer contents are changed (decremented, incremented, reset or loaded) on the clock that executes the ministep that is changing the Pointer. Therefore, the Pointer will not reflect the result until the following ministep.

Shift Done and Normalize Shift Done are available during the Shift ministep that completes the shift.

APPENDIX D
MAINTENANCE PANEL

One feature of the MLP-900 Processor of particular importance to the miniflow programmer and the service engineer is the maintenance panel controls and indicators. These facilities provide the means for initializing and operating the computer at both the miniflow and target system levels, and for performing maintenance and fault isolation. The controls and some indicators are mounted on two panels as shown in figure D-1. An additional 451 indicators, mounted on the logic cards, provide a way of monitoring the contents of processor registers and the state of certain control flip-flops. Their general layout and identification is shown in figure D-2. The function of each of the controls and indicators on the panels, and the manner in which they affect or are affected by the miniflow is described in the following paragraphs.

POWER CONTROLS

PWR ON - This pushbutton, when depressed, causes power to be applied to the processor cabinet and the associated main memory cabinet.

PWR OFF - This pushbutton, when depressed, causes power to be removed from the processor cabinet and the associated main memory cabinet.

ON - This indicator comes on when power is applied to the processor.

OVER TEMP - This indicator comes on when temperature sensors mounted on the logic cards sense that the air temperature within the processor exceeds 100°C. Power is automatically disconnected from the processor as long as this condition exists.

MASTER CLOCK CONTROLS

Master Clock - This 3 position selector switch provides control of the frequency of the processor master clock. In the NORMAL position, this will be the regular operating frequency. In the MARG position, this will be 5% higher than the normal freguency. In the AUX position, the frequency will be determined by an external generator.

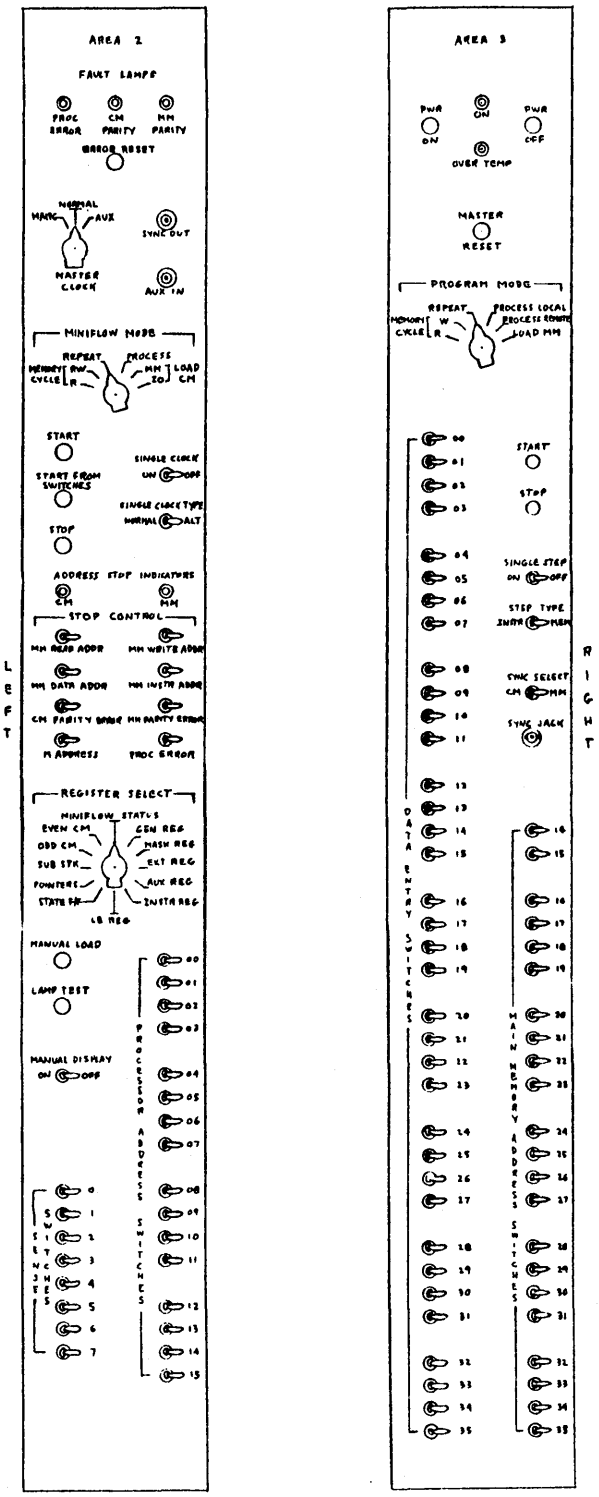


Figure D-1. MLP-900 Maintenance Panels

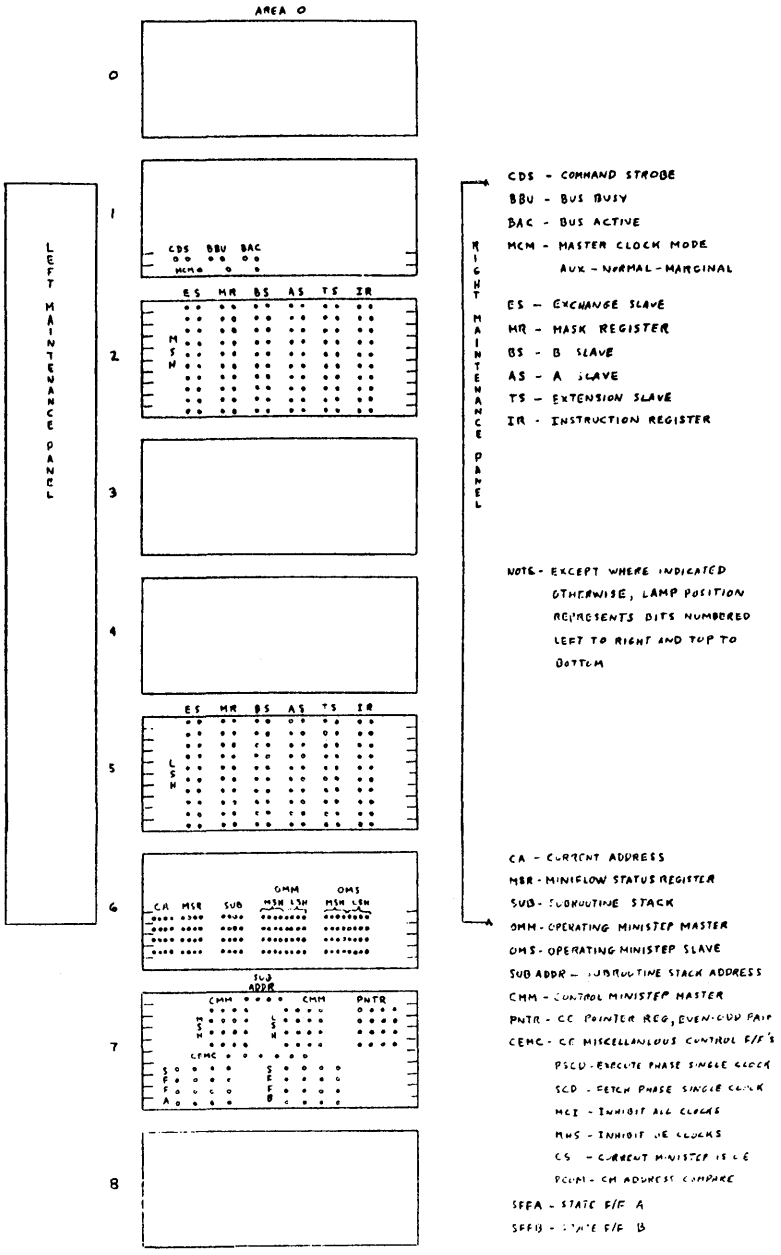


Figure D-2. MLP-900 Indicator Lamp Layout (Front View)

Sync Out - This jack provides a source of clock pulses for syncing an oscilloscope.

Aux In - This jack provides a means for connecting an external frequency generator into the master clock circuit.

MINIFLOW OPERATING CONTROLS

Miniflow Mode - This 6 position selector switch provides control of the operating mode of the miniflow level machine. In the MEMORY CYCLE modes, ministeps are not executed. Instead sequential control memory addresses are accessed beginning with the address in the Current Address register and continuing up to the address set into the Processor Address Switches. When the upper bound is reached, accessing is restarted at the address set into the lower 16 bits of the MM Address switches thus resulting in a cyclic operation. When R (Read) is selected, access to a particular address is made only once per cycle. When RW (Read-Write) is selected, however, two accesses are made; the first to read out the contents of the location, and the second to write into the location the 32 least significant bits of the Data Entry switches. In the REPEAT mode, ministeps are executed normally except when one of the pair is accessed from a CM address equal to that set into the Processor Address switches. When this occurs neither ministep is executed and a branch is forced to the CM address set into the MM Address switches. In the PROCESS mode, ministeps are executed normally. In the LOAD CM modes, ministeps are executed normally, except that when operation is first started, entry is forced to the starting location of the LOAD CM miniflow. This is FF06 if loading is done from MM, and FF04 if loading is done from IO.*

Start - This pushbutton, when depressed, turns on the clock control flip-flop (CKC) which causes the miniflow level machine to begin operation. It also causes a special address entry for the Load CM modes if the clock control flip-flop is off. In the other modes, operation begins at the continuation address.

*Loading from IO is not implemented at this time.

Start From Switches - This pushbutton, when depressed, also turns on the clock control flip-flop, which causes the miniflow level machine to begin operation, but entry is forced to the address in the Processor Address switches if the clock control flip-flop is off.

Stop - This pushbutton, when depressed, turns off the clock control flip-flop and thus stops operation at the miniflow level.

Master Reset - When this pushbutton is depressed, the Miniflow Status register, Pointer registers 0-7, State F/F registers 0-15 in group 0 and 0-7 in group 1, and all the External Bus control flip-flops will be cleared. Several other special control flip-flops not accessible to the miniflow programmer are also cleared to assure a viable initial state.

Single Clock - When this toggle switch is set to the ON position, only one clock pulse occurs when the Start pushbutton is depressed.

Single Clock Type - When this toggle switch is set to the NORMAL position, one or two ministeps will be executed each time the Start pushbutton is depressed. When set to the ALT (Alternate) position, execution occurs every other depression of the Start pushbutton. Ministeps are thus current for 2 clock periods which allows the results of their execution to be observed.

Sense Switches - These eight toggle switches permit manual selection of programming options at the miniflow level. The state of these switches are sensable by the miniflow as State Pseudo F/F's (R12 in group 1).

TARGET LEVEL OPERATING CONTROLS

Program Mode - This six position selector switch provides control of the operating mode of the target level machine. It does this by controlling the CM entry address to which miniflow control is forced when the program Start pushbutton is depressed. These addresses are:

Memory Cycle R	65293_{10}	$FFOD_{16}$
Memory Cycle W	65293	FFOD
Repeat	65	0041
Process Remote	66	0042
Load MM	Unassigned*	

The Memory Cycle entry address is in the read-only portion of control memory. A hard miniflow starting there permits manual loading, displaying or testing of main memory. The other entry addresses except for Load MM are in the RW portion of control memory and thus require appropriate soft miniflow to be loaded before these modes can be used. Selection of the Repeat mode enables the MM Address Compare Action Request to be set when a main memory access is made to an address equal to that set into the MM Address switches. An appropriate soft miniflow routine must be provided if this feature is to be used.

The Process Local mode provides a way to control machine operation at the target level. The Process Remote mode provides a way to put the processor in a state where target level operation may be controlled from the operator console. The Load MM mode will provide a way to get main memory loaded with miniflow and/or target level programs from an IO device in systems which do not have an IO processor.

Start - This pushbutton when depressed, turns on the RUN flip-flop, and forces miniflow control to one of the control memory addresses listed above. The latter occurs only when RUN is off. It also will cause the Memory Bus Inhibit flip-flop (MBI) to be reset if it is on.

Stop - This pushbutton, when depressed, causes the RUN flip-flop to be reset and the Memory Bus Inhibit flip-flop to be set if the mode is Memory Cycle. A miniflow entry is also forced to control memory address 64 if not in the Memory Cycle modes.

Single Step - This toggle switch, when set to the ON position causes the processor to operate in a step mode. In this mode, each depression of the program Start pushbutton will cause only those ministeps which are required to execute one instruction or to progress up to the next memory cycle.

*to be in read-only memory

Single Step Type - This toggle switch selects whether stepping is done on an instruction basis or on a memory cycle basis.

MM Address Switches - These 22 toggle switches allow main memory addresses to be selected for starting, address stopping or defining the upper bounds for the target level Memory Cycle and Repeat modes, or the lower bounds for the miniflow level Memory Cycle and Repeat modes.

FAULT INDICATION

Proc Error - This indicator comes on whenever a register parity or compare error is detected in the processor.

CM Parity - This indicator comes on whenever a word read out of the even or odd control memory has a parity error.

MM Parity - This indicator comes on whenever a word read in from an external bus contains a parity error.

Error Reset - This pushbutton, when depressed, will cause any of the three fault indicators which are on to go off.

BREAKPOINT FACILITIES

Eight stop switches are provided which permit the processor to be stopped when any of several different kinds of events occur. Four of these cause a halt at the target system level when a main memory address is accessed in a certain way. In these cases, restarting is accomplished by depressing the program Start pushbutton. To cause a halt at least two of these four switches must be set - one of the read or write stop switches and one of the data or instr stop switches. The other four switches cause a halt at the miniflow level (i.e. the clock is turned off) which requires that the miniflow Start pushbutton be depressed to restart operation. Two stop indicators plus the three fault indicators permit the event causing the halt to be determined where more than one type of stopping event has been selected.

CM stop - This indicator comes on when a processor halt occurs at the miniflow level. It will go out when miniflow Start is depressed.

MM stop - This indicator comes on when a processor halt occurs at the target system level. It will go out when program Start is depressed.

MM Read Addr. - When this toggle switch is on (to the right) a target level halt will occur after a main memory read operation has occurred at the address set into the MM Address switches.

MM Write Addr. - When this toggle switch is on, a target level halt will occur after a main memory write operation has occurred at the address set into the MM Address switches.

MM Data Addr. - When this toggle switch is on, a target level halt will occur after data has been read from or written into the main memory address set into the MM Address switches.

MM Instr Addr. - When this toggle switch is on, a target level halt will occur after an instruction has been read from the main memory address, set into the MM address switches, and executed.

MM Parity Error When this toggle switch is on, a miniflow level halt will occur after a data or instruction word has been read from main memory with a parity error.

CM Parity Error When this toggle switch is on, a miniflow level halt will occur after a ministep has been read from control memory with a parity error.

Proc Error - When this toggle switch is on, a miniflow level halt will occur after a processor compare or register parity error has occurred or an invalid control memory address has been made.

CM Address - When this toggle switch is on, a miniflow level halt will occur after a ministep has been read from the control memory location set into the Processor Address switches.

MANUAL LOAD AND DISPLAY FACILITIES

Register Select This 12 position selector switch permits selection of the register group which is to be loaded or displayed.

Processor Addresses Switches — These 16 toggle switches are used to address the particular register within a group which is to be loaded or displayed. Where less than 16 are required, those effective will always be on the least significant end. These switches are also used for the CM Address stop function, with the Start From Switches pushbutton, and they may be loaded into the processor with the GENT ministep. The '1' position is to the right.

Data Entry Switches - These 36 toggle switches provide a source for data to be loaded into the processor. Correct parity is generated automatically. The '1' position is to the right.

Manual Load - This pushbutton, when depressed, causes the register addressed by the Register Select switch and the Processor Address switches to be loaded with the data set into the Data Entry switches. The pushbutton is active only when the processor is stopped at the miniflow level.

Manual Display - This toggle switch, when set to the ON position causes the contents of the register addressed by the Register Select switch and the Processor Address switches to be displayed on the appropriate set of indicator lamps on the logic cards. The manually selected display occurs only when the processor is stopped at the miniflow level.

Lamp Test - This pushbutton, when depressed, causes all non-defective indicators mounted on the logic cards to be turned on.

SCOPE SYNC FACILITIES

Sync Jack - This jack facilitates connecting an address compare sync signal to a scope for synchronization purposes.

Sync Select - This toggle switch permits selecting either a sync signal which occurs when the CM address matches that set into the Processor Address switches, or one which occurs when the MM address matches that set into the MM Address switches.

APPENDIX E
MINISTEP FIELD DECODES

E.1 MAJOR OPERATION CODE (BITS 0-3)

CODE	MINISTEP
0000	GEAR
0001	CEDE
0010	SHIN
0011	CHAL
0100	GENT
0101	TEXT
0110	Undefined
0111	CHAR
1000	BRAT
1001	BENT
1010	BORE
1011	BRAD
1100	BEAD
1101	BLOT
1110	MAST
1111	MOVE

E.2 GEAR FIELD DECODES

ARITHMETIC CODE (BITS 4-7)

0000	*	LOGICAL:	$\overline{A}$ or B
0001		LOGICAL:	A and B
0010		LOGICAL:	A ← B
0011		LOGICAL:	A and $\overline{B}$
0100		LOGICAL:	A or $\overline{B}$
0101		LOGICAL:	A and B
0110		LOGICAL:	A or B
0111		LOGICAL:	A ← $\overline{B}$
1000		LOGICAL:	A Exclusive OR B
1001		BINARY:	A + B
1010		BINARY:	$\overline{A}$ + B + 1
1011		BINARY:	A + B + COF1
1100		BINARY:	A + $\overline{B}$ + COF1
1101		BINARY:	$\overline{A}$ + B + COF1
1110		BINARY:	A + $\overline{B}$ + 1
1111		LOGICAL:	A Exclusive OR $\overline{B}$

* $\overline{X}$ implies the one's complement of X

MASK ADRS (BITS 8-11)

0000	Mask 0 or 16 (Controlled by Mask Bank Select, MBS)
0001	Mask 1 or 17
-	
-	
-	
1111	Mask 15 or 31

SHIFT AMOUNT (BITS 12-15)

	Right		Left
0000	0	1000	0
0001	1	1001	1
0010	2	1010	2
0011	4	1011	4
0100	6	1100	6
0101	8	1101	8
0110	12	1110	12
0111	16	1111	16

CLEAR (BIT 16)

0	NoClear
1	Clear

TEST (BIT 17)

0	Result to A
1	A Unchanged

IA (BIT 18)

0	Bits 19-23 are OP.A
1	Bits 19-22 are IND. ADRS and Bit 23 is OR Bit

OP.A (BITS 19-23)

00000	G0	IA = 0
00001	G1	
-		
-		
-		
11111	G31	

IND. ADRS (BITS 19-22)

0000	P0	Used as indirect pointer when IA = 1
0001	P1	
-		
-		
-		
1111	P15	

OR (BIT 23)*

0	Normal Indirection
1	Force Odd Address

*. IA must be a 1

B SEL (BITS 24-25)

00	General Registers (direct or indirect)
01	Pointer Registers
10	Short Immediate
11	Long Immediate

IB (BIT 26)*

0 Bits 27-31 are OP.B
1 Bits 27-30 are IND. ADRS and Bit 31 is OR Bit
* B SEL must be a 1

OP. B (BITS 27-31)*

00000	G0
00001	G1
-	
-	
-	
11111	G31

* IB and B SEL must both be zero

IND. ADRS (BITS 27-30)

0000	P0	Used as indirect pointer when IB=1 and B SEL = 0
0001	P1	Used as 8 bits of data from the Pointer Register (leading zeros) when B SEL = 01.
-		
-		
-		
1111	P15	

OR (BIT 31)*

0 Normal Indirection
1 Forces Odd Address
* IB must be one and B SEL must be zero

SHORT IM. (BITS 26-31)

When B SEL =10 this field is used as the low order 6 bits of the B Operand. 30 high order zeros are created.

LONG IM. EXT. (BITS 28-31)

When B SEL =11 this field is used as the high order 4 bits of the B Operand. For explanation of the low order 32 bits see LONG IM. (BITS 0-31).

LONG IM. (BITS 0-31)

A word of Control Memory is used as a Long Immediate word (low order 32 bits) when the previous word was a ministep specifying Long Immediate (B SEL = 3 in a GEAR, CEDE, SHIN).

E.3 CEDE FIELD DECODES

EXCHANGE CODE (BITS 4-7)

0000	FIN	Command Type
0001	WIN	Combinatorial Type
0010	FOP	Command Type
0011	SOP	Command Type
0100	GAD	Command Type
0101	RMW	Command Type
0110	WSS	Data Type
0111	WIF	Combinatorial Type
1000	WOF	Combinatorial Type
1001	WAS	Data Type
1010	Undefined	- -
1011	Undefined	- -
1100	WOP	Data Type
1101	Undefined	- -
1110	ROW	Command Type
1111	WON	Combinatorial Type

CONDITION CODE (BITS 9-11)

These 3 bits are placed unchanged into bits 2-4 of the 36 bit command word where they are externally decoded.

OP.A EXTEND (BITS 9-11)*

These 3 bits concatenate with the 5 bits in the OP.A field to create an 8 bit address.

* IA must be zero

OP.A GROUP (BITS 12-15)

0000	General Registers
0001	Data Mask Registers
0010	Miscellaneous (See OP.A)
0011	Unassigned
0100	Auxiliary Register Bank 0
0101	Auxiliary Register Bank 1
0110	Auxiliary Register Bank 2
0111	Auxiliary Register Bank 3
1000	Control Engine
1001	Control Engine
1010	Control Engine
1011	Control Engine
1100	OE Language Board Register Bank 0

OP.A GROUP (BITS 12-15) con't.

1101	OE Language Board Register Bank 1
1110	OE Language Board Register Bank 2
1111	OE Language Board Register Bank 3

SUB (BIT 16)*

0	Add B Operand
1	Subtract B Operand

* FIN, WIN, FOP, WIF, SOP, GAD, RMW Exchange Codes only.

OP.A (BITS 19-23)*

These 5 bits (along with OP.A EXTEND) address sequentially through the OP.A GROUP specified.

The sequence for the Miscellaneous Group is:

0000	Data Entry Switches
0001	Main Memory Address Switches
0010	Processor Address Switches
0011	Unused
0100	Primary Instruction Register
0101	Secondary Instruction Register
0110	Unused
0111	Unused
1000	External Register In 0
1001	External Register In 1
1010	External Register In 2
1011	External Register In 3
1100	External Register Out 0
1101	External Register Out 1
1110	External Register Out 2
1111	External Register Out 3

* IA must be zero

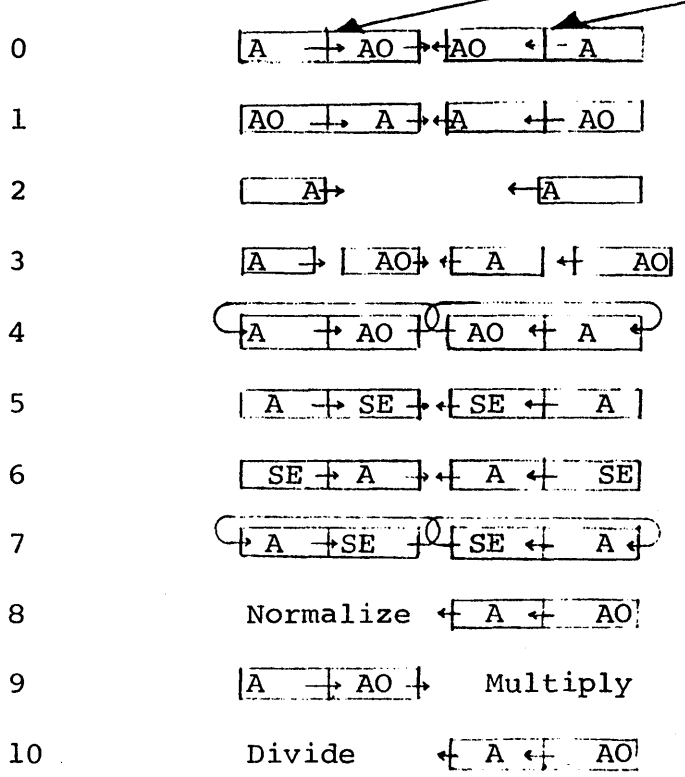
TEST, IA, IND. ADRS, OR, B SEL, IB, OP.B, SHORT IM., LONG IM. EXT.

Same as GEAR FIELD DECODES

E.4 SHIN FIELD DECODES

SHIFT CODE (BITS 4-7)

SHIFT AMOUNT IS:
0XXX (Right) 1XXX (Left)



11 Unassigned

12 Unassigned

13 Unassigned

14 Unassigned

15 Unassigned

NOTES: A is register addressed by the A address of the SHIN. AO is the same address with a 1 ORed into the low order bit position forcing an odd address. SE is the Shift Extension register, G31.

MASK ADRS (BITS 8-11)

Same as GEAR

SHIFT AMOUNT (BITS 12-15)

Same as GEAR

IS (BIT 16)

0	Shift amount is specified in Ministep
1	Shift amount is specified in P7 (Indirect)

TEST, A ADRS (Composite), B ADRS (Composite)

Same as GEAR

E.5 CHAL AND CHAR FIELD DECODES

CHAL and CHAR ARITH CODE (BITS 4-7)

0000	*DECIMAL: A + B
0001	DECIMAL: A + B + COF1
0010	Undefined
0011	Undefined
0100	DECIMAL: A + $\hat{B}$ + 1
0101	DECIMAL: A + $\hat{B}$ + COF1
0110	LOGICAL: A And B
0111	LOGICAL: A$\leftarrow$B
1000	BINARY: A + B
1001	BINARY: A + B + COF1
1010	BINARY: $\overline{A}$ + B + 1
1011	BINARY: $\overline{A}$ + B + COF1
1100	BINARY: A + $\overline{B}$ + 1
1101	BINARY: A + $\overline{B}$ + COF1
1110	LOGICAL: A OR B
1111	LOGICAL: A EXCLUSIVE OR B

* + implies decimal or binary add as indicated
$\overline{X}$ implies the one's complement of X
$\hat{X}$ implies the nine's complement of X

MASK ADRS (BITS 8-11)

Same as GEAR (only last 8 bits of Mask are active)

BYTE A (BITS 12-13)*

00	BYTE 0
01	BYTE 1
10	BYTE 2
11	BYTE 3

* IND A BYTE must be a 0

BYTE B (BITS 14-15)*

00	BYTE 0
01	BYTE 1
10	BYTE 2
11	BYTE 3

* B SEL must be 00, 01, or 10

IND A BYTE (BIT 16)

0	Byte number for A operand is specified in the Ministep
1	Byte number for A operand is specified by Pointer 00 (Indirect)

TEST (BIT 17)

0	Result to A (Modify Pointers 00-03 when Byte is indirect)
1	A Unchanged (No modification of Pointers)

B SEL (BITS 24-25)

00	General Registers (direct B byte)
01	Pointer Registers
10	Character Immediate (Bits 14-15, 26-31)
11	General Registers (indirect B byte specified by Pointer 01)

IMMED CHAR (BITS 14-15, 26-31)*

These 8 bits are used as the B operand input to the Decimal Adder.
* B SEL must be

IA, OP.A, IND. ADRS, OR, IB, OPB

Same as GEAR

E.6 GENT FIELD DECODES

TO/FROM (BIT 5)

0	Operand A is the TO address
1	Operand A is the FROM address

PAR 1/PAR 0 (BITS 6-7)

00	Normal Parity
01	Forced 0 Parity
10	Forced 1 Parity
11	Forced 1 Parity

OP.B GRP (BITS 16-17)

00	General Registers
01	Mask Registers
10	Miscellaneous (See CEDE, OP.A (BITS 19-23)
11	Control Engine

OP.B (BITS 27-31)

These 5 bits address sequentially through the OP.B GRP specified.

OP.A EXTEND, OP.A GROUP, IA, OP.A, IND ADRS, OR,IB, B SEL

Same as CEDE

E.7 BRAT, BENT AND BORE FIELD DECODES

TEST MODE (BITS 4-5)

00	Test A and Move A to B
01	Test A OR B
10	Test A AND B
11	Test A EXCLUSIVE OR B

$A/\overline{A}$ (BIT 6)

0	Use $\overline{A}$
1	Use A

$B/\overline{B}$ (BIT 7)

0	Use $\overline{B}$
1	Use B

TEST BIT A (BITS 8-15)

Bits 8 - 11	Register address within a CE Group
Bits 12 - 14	Bit address within a CE Register
Bit 15	CE Group 0 or 1

TEST BIT B (BITS 16-23)

Bits 16 - 19	Register address within a CE Group
Bits 20 - 22	Bit address within a CE Register
Bit 23	CE Group 0 or 1

RELATIVE ADDRESS (BITS 24-31)

This field specifies the amount to be added to the continuation address to obtain the effective address to be used if branching occurs. Negative amounts must be expressed in 2's complement form. The amount in the RelativeAddress field will be construed as a negative number in 2's complement form if bit 24 is a one.

E.8 BRAD FIELD DECODES

POINTER (BITS 8-11)

This field sequentially addresses the CE Pointer Registers

MODIFIER (Bits 12-15)

This field specifies the value in two's complement form, for Pointer Register modification.

B/$\overline{B}$, TEST BIT B, RELATIVE ADDRESS

Same as BRAT

E.9 BEAD FIELD DECODES

TEST MODE (BITS 4-5)

00	Conditional Branch (Absolute Address)
01	Branch plus Pointer (Absolute Address)
10	Continuation plus Pointer
11	Conditional Branch (Relative Address)

A/$\overline{A}$ (Bit 6)

Same as BRAT

ENTER (BIT 7)

0	No Subroutine entry
1	Subroutine entry is performed by storing the Continuation Address into the Subroutine Return Register addressed by the Subroutine Stack Pointer and incrementing the Pointer by 1.

POINTER (BITS 8-11)*

Same as BRAD

* TEST MODE must be 01 or 10

TEST BIT A (BITS 8-15)*

Same as BRAT

* TEST MODE must be 00 or 11

EXTENDED BRANCH ADDRESS (BITS 16-31)*

This field specifies the Absolute Address to be used for branching or the amount to be added to the Continuation Address to obtain the effective address for Relative branching. Negative relative branch amounts must be expressed in 2's complement form.

* TEST MODE must be 00, o1 or 11.

E.10 BLOT FIELD DECODES

STREAM (BIT 4)*

0	No function
1	If this Ministep is coupled with a CEDE/WOP the "Bus Busy" State flip-flop will be prevented from being reset until Pointer 01 reaches a value of 1.

* To utilize this function the memory or peripheral device must be equipped to accept streams of data words with no intervening command words.

BLOT CODE (BITS 5-7)

000	RCM	Read from CM to OE
001	WCM	Write to CM from OE
010	RSB	Read from Subroutine Stack to OE
011	WSB	Write to Subroutine Stack from OE
100	MOE	Move within OE
101	WBP	Write to CM from OE with Bad Parity
110	LMB	Load Multiple Blocks
111	Undefined	

RELATIVE ADDRESS (BITS 24-31)

Same as BRAT although the condition for branching is not the same. The condition for branching is that Pointer Register 01 not have the value 1. However, an exception occurs when executing LMB. If the chain bit is true a branch will occur even though Pointer 01 is 1. In all other cases the Continuation Address is taken with the one exception occurring when the LMB causes the Current Address Register to be loaded.

E.11 MAST FIELD DECODES

LOGIC CODE (BIT 4-5)

00	If B then RESULT←A
01	RESULT←A OR B
10	RESULT←A AND B
11	RESULT←A EXCLUSIVE OR B

A/$\overline{A}$ and B/$\overline{B}$

Same as BRAT

STATUS BIT A

Same as BRAT, TEST BIT A

STATUS BIT B

Same as BRAT, TEST BIT B

RESULT STATUS BIT (BITS 24-31)

Bits 24-27	Register address within a CE Group
Bits 28-30	Bit address within a CE Register
Bit 31	CE Group 0 or 1

E.12 MOVE FIELD DECODES

MOVE CODE (BITS 4-7)

0000 MSI Move Short Immediate
0001 MOM Move One Status Bit to Many (fill register)
0010 MAR Move And Replace
0011 MAC Move And Complement
0100 MCL Move And Clear Masked Bits
0101 MDB Move Double Byte (i.e. 16 bit move)
0110 Undefined
-
-
-
1111 Undefined

FROM ADDRESS (BITS 8-15)*

Bits 8-11 Register address within a CE Group
Bits 12-15 CE Group
* For MOVE CODE MOM the FROM ADDRESS is the same as BRAT, TEST BIT A

TO ADDRESS (BITS 16-23)

Bits 16-19 Register address within a CE Group
Bits 20-23 CE Group

IMMEDIATE MASK (BITS 24-31)*

These 8 bits form an immediate mask for all CE moves except MOVE CODE 0101 MDB.
* The Subroutine Return Registers are not maskable.

GLOSSARY OF COMMONLY USED TERMS

TARGET LEVEL TERM	COMPARABLE MICROPROGRAM LEVEL TERM
Software	Firmware (Microprogram)
Instruction Set (Order Code)	MINIFLOW Language
Instruction	Ministep
Main Memory	Control (Microprogram) Memory
Program Execution	Emulation or Interpretive Execution
Interrupt	Action Request
Program Status Word	MINIFLOW Status Word
Instruction (Program) Counter	Current Address Register
Index Registers	Pointer Registers
Mode Control (Condition Code) Elements	State Flip-Flops
Memory Fetch/Store Functions	Data Exchange Operations
I/O Operations	Transfer External Ministep
Memory Protection and Relocation	Operating Engine Language Board (Address Modification)
High Speed Buffer	Auxiliary Registers

Standard

Standard Computer Corporation
1411 W. Olympic Boulevard
Los Angeles, California 90015
(213) 387-5267

777 N. First Street, Suite 600
San Jose, California 95112
(408) 294-7150

55 S. 36th Street
Boulder, Colorado 80302
(303) 428-0529

7718 Tanglecrest Drive
Dallas, Texas 75240
(214) 239-9627

8 South Michigan Avenue
Suite 720
Chicago, Illinois 60605
(312) 726-0277

17500 Northland Park Court
Southfield, Michigan 48076
(313) 352-1710

Suite 1325 — Land Title Building
Broad & Chestnut Streets
Philadelphia, Pennsylvania 19102
(215) LO3-6350

7100 France Avenue South
Minneapolis, Minnesota 55435
(612) 926-0706

235 Bear Hill Road
Waltham, Massachusetts 02154
(617) 891-5083

2 West 45th Street
New York, New York 10036
(212) 661-1834