Harry J. Saal
May 22, 1970
CGTM #90

A BRIEF TOUR THROUGH THE MICROPROGRAMMED RESEARCH FACILITY

Stanford University (SLAC), with the Standard Computer Corporation, are preparing a new research facility. The proposed equipment is listed and shown in Figure 1. The facility is based around two processors, an MLP-900 and an IOP. Each is a microprogrammable machine where changes in the microprograms can be made in the field without any rewiring, etc. Neither processor is well known and data concerning the MLP-900 is restricted, since that CPU is currently officially unannounced. The following is an abbreviated tour through the extensive facilities provided.

MLP-900, Brief Description

The MLP-900 is a general purpose micro-programmed computer. Access to machine registers and flip-flops (switches), data operations, and sequencing is controlled by micro-code, called MINIFLOW. MINIFLOW instructions, called ministeps, have a form similar to conventional machine order codes.

The MLP-900 is organized as two relatively independent sub-processors. They are:

1) Operating Engine (ØE) - performs arithmetic and logical operations on the data and communicates with main memory over external busses. All registers for data manipulation are in the ØE.
   (The ØE is analogous to the arithmetic unit of a conventional processor).
2) Control Engine (CE) - performs status manipulation and ministep sequencing. The CE contains all status registers and control memory.

Ministeps reside in control memory, which may be implemented as read/write, read-only, or both. Control memory words are 32 bits in length, with a 16 bit address. The memory is implemented in integrated curcuit logic, with access time of less than 50 nano seconds for read. Current technical considerations permit any amount of read-only memory, and up to 5K (K=1024) of read/write memory.

Ministeps are divided into two classes - OE and CE, which are directives to the respective processors. Ministeps are fetched from control memory in a similar manner to a conventional machine -- e.e., sequentially unless a branch is taken, or a micro-program interrupt (called "action request") occurs.

Ministeps are fetched in pairs from memory. If the pair is an ØE ministep followed by a CE ministep, they are executed concurrently. Each ministep, or pair, generally takes one clock cycle (128 ns) to execute. Thus, the micro-program parallelism is limited to execution of ministep pairs, as well as whatever related operations can be combined in a single ministep. Main memory references may be overlapped with ministep execution because separate ministeps are used to send addresses to memory and to transfer data previously addressed. (Generally, the MLP-900 is an instance of a "vertical" micro-program organization.)

Facilities for supplementing the MLP-900 hardware with user-defined hardware are provided with the "language board" (LB). The LB may be used to perform address translation, data formatting, and instruction decoding. While not required for operation, use of the LB will improve efficiency and save control memory space for repetitive, frequently used operations of the types named. There is provision for up to four different LBs to be simultaneously installed, with the selection of the active LB to be micro-program controlled.

The target language instruction fetch mechanism involves an interaction between the LB and the ØE ministeps that reference main memory. The ministep "Wait for instruction" (WIN), in conjunction with the LB, does most of the work for instruction decoding. The WIN ministep will suspend ministep execution until the instruction is available from main memory; then the LB may separate the fields of the instruction, perform an address calculation, send out the address for the operand if needed, and supply a control memory branch address based on partial decoding of the opcode, thus effecting a forced branch.

Micro-program interrupt processing is also connected with the WIN ministep. There are three classes of interrupts for (1) conditions internal to the processor (e.g. parity errors), (2) external conditions (e.g. memory errors), and (3) interrupts defined by the user for the target language. Forced branches are taken for class (1) when they occur, for (2) when a wait for main memory occurs, and for (3) when a WIN is executed. Interrupt masking and priority servicing is provided.

I/O is done by a separate processor, the I/O Processor (IOP), which communicates with the MLP-900 by external interrupts (class 2). Data transmission is done directly between the IOP and main memory.

Following is a more detailed description of ØE and CE capabilities.

2. Operating Engine (ØE)

The Ø E contains registers and logical elements for data manipulation. Data transfers to and from main memory go through the ØE. The major ØE registers are:

a) 32 General registers - (36 bit data length) used by most ØE ministeps for operands and results. Ministeps may reference them directly or indirectly (reg address obtained from a CE register).

b) 32 Data mask registers - (36 bit data length) used by ØE ministeps to supply operand masks so that arithmetic and logical operations may be performed on data fields less than 36 bits in length.

c) 0-1024 Auxiliary register - (36 bit data length) high speed intermediate storage registers. Data may be transferred between the auxiliary registers and the ØE registers.

d) 0-1024 ØE Language board registers - (36 bits) - these addresses, if defined on the language board, may be used to transfer data between the language boards and the ØE.

e) Primary and secondary instruction registers - (36 bits) - used by the language board during instruction fetch operations.

f) External registers - (36 bits) - external bus input and output registers - generally referenced implicitly by ministeps performing external data transfers.

ØE ministeps fall into the following classes:

a) General arithmetic (GEAR) - perform parallel binary 36 bit arithmetic and logical operations. Use of the mask register permits operation on fields shorter than 36 bits, which need not be composed of contiguous subfields. The results of the operation may be shifted also if desired. 2's complement representation of negative numbers is assumed. Carry treatment is ministep controlled. 16 different logical operations on the two operands are provided.

b) Character/Decimal (CHAR and CHAL) - decimal arithmetic and binary arithmetic/logical operation on 8-bit bytes. The bytes are hardware defined as bits 4-11, 12-19, 20-27, 28-35 of the 36 bit word. Decimal arithmetic uses 10's complement representation of negative numbers, where the digits are stored in 4-bit BCD coding (2 per byte).

Binary operations are similar to GEAR. Masking operands, but not shifting of results, is provided. CHAL and CHAR ministeps may specify the byte within the operand word explicitly or indirectly. In the latter case, facilities for automatically counting field lengths and byte positions are provided. 14 different functions of the two operand bytes are provided.

c) Shift (SHIN) - perform single and double length shifts - either simple or circular. Shifts of 1, 2, 4, 6, 8, 12, or 16 bits may be done in one clock cycle. Facilities are provided for specification of arbitrary shift with a 2-instruction loop. Also included are multiply and divide instructions, and a special normalize shift that derives its shift amount from the CE language board for floating point normalization. 11 shin ministeps are available.

d) General data transfer (GENT) - directs transfer of data between ØE registers or an ØE register and the CE data bus (where it is received or has been sent by a CE ministep).

e) Conditional external data exchange (CEDE) - these ministeps initiate and terminate transfers between the ØE and main memory. Generally one CEDE operation sends out an address. At a later time, to allow overlap, another type of CEDE is used to transmit/receive the data, causing a wait to occur if the memory is not yet ready. The ØE language board can be invoked during CEDE operation for operations such as address translation and/or relocation, or data reformatting or instruction decoding.

3) Control Engine

The CE contains control memory, various registers for recording the status of the MLP-900 or the target machine, and registers for sequence control. The major registers are:

a) 256 state flip/flops (SFF) - These are 1-bit registers used for recording MLP-900 status (conditions produced by ØE ministeps - e.g. carry-out of adder, hardware exceptions), communication with the language boards, or saving information about the target language machine status. Some of the registers are "pseudo-flip flops", i.e., they are synthesized from internal signals - thus, they can be read but not set (carry-out is an example).

b) 16 pointer registers - These 8-bit registers are used by the CE for branch indexing and stack control, and by the ØE for indirect operand addressing,operands, and counting. 8 of the pointer registers are "pseudo-registers" - they are actually outputs from the language boards, and can be read but not set or counted.

c) Subroutine stack - Sixteen 16-bit registers to save return address for miniflow subroutine - Stack manipulation and checking is handled automatically.

d) Miniflow status word - contains current miniflow address, interrupt mask, and language board selection bits.

CE ministeps perform one of several functions:

a) Conditional branching or subroutine entry - test for logical combinations of 2 state-flip flops to determine if branch is to be taken. One form also increments or decrements a pointer register, another form uses the pointer register as a branch address index. Branch addresses may be relative (-128 to +127) or absolute.

b) Manipulation of status flip-flops - perform logical operations on SFF.

c) Transfer data between CE registers or between CE register and ØE data bus.

d) Block transfer - transfers blocks of data between CE and ØE registers, control memory, and main memory. Block transfers are used to load control memory from main memory and to initialize or save MLP-900 registers.

IOP - Brief Description

The IOP is a considerably simpler and less flexible processor than the MLP-900. It was designed for use in earlier systems produced by Standard Computer Corporation and strongly reflects certain objectives for those IBM 7044/7094-like systems. We expect to replace the present IOP with either a slower, less expensive "MLP" or by a multiplex processor currently being discussed, planned and designed. Since the IOP is so much simpler than the MLP-900, some more detail can be shown clearly.

The IOP is an 18-bit computer containing 4 independent, autonomous channels. Instructions are fetched two at a time from a 1K 36 bit word core memory with 1 μsec cycle time. Execution of an instruction takes 250 nsec. The IOP and MLP can communicate by passing 36 bits of information immediately through a "DXU" mailbox, as well as through common accesses of main memory. The IOP can interrupt the CPU; however, the CPUs and Channels cannot directly interrupt the IOP. They transmit basic control signals by setting bits in the channel Interrupt registers. The IOP must periodically read out the contents of these registers and test to see if any of the interrupt conditions have been set on.

There are 3 18-bit accumulators and 8 high speed scratch memory locations in addition to a shift counter, 4 level subroutine address stack and other internal facilities.

The instructions are divided into four general classes:

ENGINE CLASS
DATA TRANSMISSION CLASS
SINGLE OPERAND CLASS
PROGRAM CONTROL CLASS.

These four classes will each be described in turn.

Sequential program control is controlled by four ten-bit registers, called Program Counters (PC0, PC1, PC2, and PC3). During execution, the inner computer is operating at one of the four program levels; the corresponding PC is updated by one after each instruction pair is fetched. Thus, the PC points to the word address of the next sequential instruction pair.

The contents of the active PC may be otherwise modified by Transfer type instructions, which affect program branching.

Certain types of transfer instructions allow the inner computer to transfer to a different level. On a "descend and transfer" instruction, the machine loads a transfer address into the PC at the next lowest level, and then begins execution at that level; thus, it branches to a subroutine. (Level 0 is the highest level; level 3 is the lowest). On an "ascend" transfer, it begins execution at the next higher level (or a level 0 if at level 3); thus, it returns from a subroutine, as the next sequential instruction location was retained in the higher PC.

The levels "wrap-around"; thus, an "ascend and transfer" executed in level 3 will cause the machine to begin at level 0.

ENGINE CLASS

The Engine Class is divided into two sub-classes as follows:

Scratch Memory Operand

Immediate Operand

In the Engine Class, the Engine combines two operands. Operand A is one of the three Accumulators. Operand B is "Immediate Data", taken from bits 6 through 17 of the instruction itself, or the data from one of the SM registers. The output of the Engine may be stored into the specified AC.

Bits 1, 2 specify the Engine function: ADD,AND,OR and PASS Operand B.

Bits 4,5 specify the AC(s) which are used as data and into which the Engine output is loaded (All three ACs, AC1, AC2, AC3). The only valid specification of "All three ACs" is with the "Pass Operand B" engine function. This is used when it is desired to place the data of Operand B into all three ACs with one instruction. Other uses of this specification are invalid.

If bit 3=0, the output of the Engine will not be loaded into the specified AC(s), but testable flip-flops will be set.

On Immediate instructions, the Internal PRIs (Previous Result Indicators) are set. On Scratch Memory instructions, bit 8 specifies which pair of PRIs will be set by the Engine operation (internal or target).

Within a pair, the Zero PRI will be set on at the end of the instruction only if the engine output was all zeros. The Carry Not PRI will be set on only at the end of an ADD function, and only if there was no high-order carry out of bit position 0 of the engine.

DATA TRANSMISSION CLASS

The Data Transmission Class is used to transmit data between one or two ACs and a main memory, a control memory or a SM register. The class is divided into three subclasses as follows:

Scratch Memory

Memory, Direct Address

Memory, Indirect Address

Scratch Memory

Eighteen bits of data are sent from a specified AC to a specified SM register. The contents of the ACs are not modified. The AC is specified by bits 4,5 of the instruction: (invalid,AC1, AC2, AC3).

There are three special cases of Scratch Memory sub-class, with AC=00. These special instructions allow the inner computer to halt, and to interrupt and awake the MLP-900.

Memory Direct

AC1 and AC2 are treated as one 36-bit register. Data is moved between the register and a control memory cell, a main memory cell, or a channel register.

Bit 3 controls the direction of data transmission:(Write or read).

Bits 8 through 17 specify the direct address of a memory word or channel register.

Bits 4,5 of the instruction and bit 15 of the Active Processor/Channel Register control what memory is accessed:

a) CPU Control Memory

This format allows the inner computer to access data in the control memory of one of the Central Processing Units in the system.

b) Supervisory Processor Control Memory

This format allows the inner computer to access data in its control memory.

c) Supervisory Processor Control Memory Indirect

This format allows the inner computer to access data in its control memory, with an indirect address from a SM register.

d) Control Memory or Channel Register

The address is direct from instruction bits 8-17, modified by having bits 16,17 of the Active Processor/Channel Register OR'ed in. If the direct address is greater than 0077_8, the access is to a control

memory cell. If the direct address is less than or equal to 0077_8, the access is to a Channel Register.

Main Memory, Indirect

AC1 and AC2 are treated as one 36-bit register. Data is moved between the register and a word of the Main Memory Units attached to a CPU.

Bit 8 controls the direction of data transmission : (Write or read).

The memory address is indirect through a SM register.

SINGLE OPERAND CLASS

This class of instructions operates on data in one or more AC(s) only. The only other register involved is the Shift Counter (SC). The Engine is not used, and the Internal and Target Indicators are not affected. It is divided into four sub-classes: Shift, Rotate, Normalize, Complement. In all sub-classes, except Complement, the Shift Counter is active in controlling the number of bit positions shifted.

On Shift and Rotate, the Shift Counter starts out with a shift count value. This data is either that which previously resided in the SC, or the SC is initialized directly by the data from instruction bits 11-17. When the instruction execution is complete, the value in the SC is zero.

In Normalize, the SC starts out with an initial value also. The shift direction is left. The value of the SC is incremented by one for each bit position shifted. As soon as a "1" bit is shifted out of the MSB of AC1, the shift terminates, and SC contains the incremented value.

Shift

Bits 4,5 specify the accumulator(s) (All three ACs, treated as one continuous 54-bit register, AC1, AC2, AC3).

Bit 3 controls the direction of shifting: (left or right).

Bit 10 specifies the source of the shift count: (SC is initialized with instruction bits 11-17 or SC uses previously loaded value.)

Bits shifted out of the end of a register are lost. Zeros are shifted into the register.

Rotate

AC1 and AC2 are treated as one continuous 36-bit register. The shift direction is left. Bits shifted out of bit position 0 of AC1 are shifted into bit position 17 of AC2; no bits are lost. Instruction bits 8-17 are used as in Shift above.

Normalize

ACl is shifted left until a "1" bit is shifted out of bit position 0. Instruction bits 8-17 are used as in Shift above. The attempt to normalize when the contents of ACl are all zeros will result in infinite instruction execution.

PROGRAM CONTROL CLASS

This class of instructions permits conditional and unconditional program b ranching and program level changing.

A test of one pair of the PRIs is made, and the result of the test determines what type of transfer occurs.

Bits 1,2 specify what PRI is tested:

00 = Null test (a virtual indicator which is always reset).

01 = Carry Not

10 = Zero

11 = Logical OR of Carry Not and Zero (a virtual indicator which is the logical OR of the two).

Bit 3 specifies which condition of the indicator is being tested:

0 = Condition to be met is indicator off (reset).

1 = Condition to be met is indicator on (set).

Bits 4,5 specify what action occurs if the condition is met, or not met:

Condition Met

00, 10, 11 = Transfer in level. Load current PC with transfer address.

01 = Descend and transfer. Enter next lower program level and load that PC with transfer address.

Condition Not Met

00 = No operation. Instruction does nothing. Sequential instruction execution continues.

01 = Descend. Enter next lower program level, but retain previous values of all PCs.

10 = Ascend. Enter next higher program level, but retain previous values of all PCs.

11 = Ascend to Zero. Enter highest program level (0), but retain previous values of all PCs.

Bit 6 specifies direct or indirect addressing through an SM register.

Bit 8 specifies which pair of PRIs is tested (only when bit 6 = 1): (internal or target PRIs). (If bit 6 = 0, but 8 is part of the transfer address, and the Internal PRIs are tested).

The operation and programming of the channels will not be mentioned here in any detail. They are basically controlled by 16 special registers for each channel which have been assigned special functions (channel program counter, status, interrupts, etc.).

One interesting feature will be mentioned,however. Each channel has a register known as "search memory" which is actually an 8 x 36 bit word storage. Whenever data is written into SEM, the last three bits address one of 8 sub-words for storage. The first 30 bits are then stored away in the location addressed. The first twelve of these bits constitute a search key definition for a parallel pattern matcher which operates whenever a pseudo-channel command is loaded into the "directive register" and the channel is started up. Bits 21 to 30 are set to actuate the channel functions that the hardware recognizes (i.e. count control, record control, transfer in channel, execute, etc.).

The 12 bit search key in each of the eight words is matched against six bits of the directive register. For each bit, two bits in every search word determine if a pattern match was successful. The search key decoding is:

00 don't care
01 match on zero
10 match on one
11 don't match.

If all bits of a search key match, the 10 data bits are read out. All 8 SEM words are simultaneously matching, and if several are successful, their output are OR'ed together and sent to the channel.

More detail may be found in the IOP reference manual.

EQUIPMENT LIST

1	Standard MLP-900 (3K read-write control memory)
4	Standard SC700 memories (total 128K x 40 bits)
1	Standard Input/Output Processor
1	Communications Computer (PDP8I)
1	Standard Card Reader (800 cpm)
1	Standard Card Punch (100-300 cpm)
1	IBM 2803-1 Tape Controller
2	IBM 2401 9 track Tape Drives
1	IBM 2821-2 Peripheral Controller
1	IBM 1403N1 Printer
1	IBM 2314 Disk Device (6 spindles)
1	Standard System Console
4	TTY Model 35 KSR
4	IBM 2741 Terminals

FIGURE 1: INITIAL HARDWARE CONFIGURATION