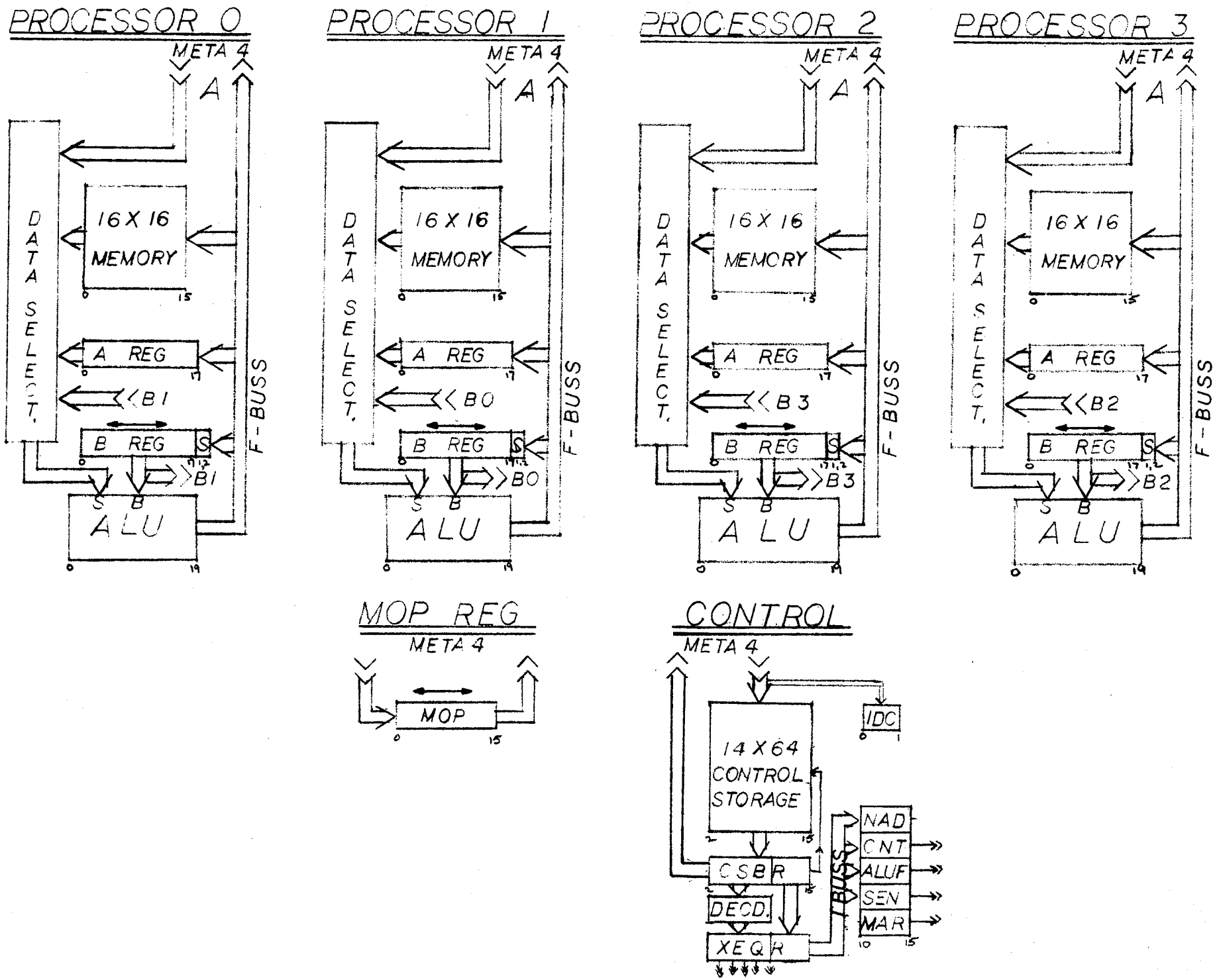
THE SIMAL E - PRINCIPLES OF OPERATION

HAROLD WEBBER JUNIOR
OCTOBER 28, 1971

SIMALE PROCESSOR - BLOCK DIAGRAM



SIMAL E - PRINCIPLES OF OPERATION

The SIMAL E (Super Integral Microprogrammed Arithmetic Logic Expediter) is a very high speed, dynamicall microprogrammed, parallel processing arithmetic unit. It was designed primarily to expide matrix operations in a graphics display processing system based on the META 4 computer. The SIMAL E is, however, sufficiently general to perform other special purpose functions either as an integral part of another system, or at different times in the same system. This generality is attributable to the SIMAL E's high speed writable control storage which, when loaded by the supervisory computer , defines the functions the machine will perform.

I - MACHINE ARCHITECTURE:

The SIMAL E is organized into two identical pairs of processors, a single operand register (MOP), and a control unit. Each of the four processors, the MOP register, and the control unit has its own META 4 I/O register. Within the machine data paths link both processors of each pair, and lines to control supervize operations and carry information about conditions in the processors and MOP register.

A - THE PROCESSORS

Each processor is an arithmetic unit of moderate power containing two work[illegible] registers, an arithmetic/

logic unit, sixteen words of memory, and various data select networks and busses to allow transfers.

1 - THE ARITHMETIC/LOGIC UNIT (ALU)

The ALU is the most important component of each processor. It is capable of performing any one of thirty two arithmetic/logic functions upon two numbers of twenty bits each. ~~Only eight of these functions are used in the SIMAL E;~~ the function of the ALU ~~being~~ is determined by the ALU function register in control. The output of the ALU forms the F-buss which can be loaded into the registers or the memory. The F-buss also goes directly to the META 4 input register. The inputs to the ALU come from the B register and the data selector.

2 - THE B REGISTER

The B register is an 18 bit shift register which can be loaded from the F-buss. 18 bits are used to prevent overflow and graphic wraparound. ~~The register also has two right end spill bits used to determine carry in when special rounding logic is enabled.~~ The register also has logic to test it for all ones or all zeros. The B register can be zeroed in addition to being shifted or loaded. The contents of the register goes directly to the ALU and to the data selector in the other processor of the pair. Because of its flexible nature, this register is most often used during computation.

3 - THE A REGISTER

The A register is an 18 bit register loaded from the F-buss. The contents of the register goes to the data selector. This register is usually used to hold intermediate results during computations.

4 - THE DATA SELECTOR

The data selector gates one of four sources into the ALU. These sources are the META 4 output register associated with the processor, the memory, the B register of the other paired processor, and the A register. Which source is selected is determined by the control unit (see SENR).

5 - THE SCRATCH PAD MEMORY

The memory has 16,18 bit words selected by the address in the MADR register in control. The specified location may be either written into from the F-buss, or may be read out to the data selector. The memory may not be used as both a source and destination of an operation (at long cycle lengths) due to its non-latching nature. This restriction does not apply to the two registers.

B - THE MOP REGISTER

The MOP register is a shift register capable of being loaded from the ~~META 4 output register associated with it~~ DATA BUS. This register can be shifted either right or left and can have as input the condition code specified in an ~~operation (see operations 0111, and1000)~~ INSTRUCTION. The

rightmost and leftmost bits ~~are~~ CAN BE tested, while the contents of the entire register goes to the META 4. The MOP register is typically used to hold a multiplier during a matrix row multiply instruction, but ~~could~~ CAN be used to return conditions to the META 4 after execution.

C - THE CONTROL UNIT

The control unit has the task of fetching micro operations from control storage and executing them; effecting changes in the processors and MOP register, and thus emulating an instruction. The control unit "pipelines" micro-operations in the sense that during any one cycle it is at once fetching an operation from control storage, decoding a second, and executing a third. This allows the SIMALE to attain great speed without need for tricky timing-dependent circuits in its design. Each stage of the process is buffered from the next by a register so that the machine can be run at a speed very much slower for debugging purposes.

1 - CONTROL STORAGE

Control storage is a ~~64~~ 128 word by ~~14~~ 15 bit high speed writable memory. A word is addressed by the low order ~~six~~ 7 bits of the Control storage buffer register. These six bits are refered to as the ~~CADR~~ CSAR ~~(control address register)~~ CONTROL STORAGE ADDRESS REGISTER. Control storage can only be written into by the META 4 so to the SIMAL E control storage ~~is~~ APPEARS TO BE THEREFORE, read only and cannot be used for storage of operands.

2 - CONTROL STORAGE BUFFER REGISTER (CSBR)

This 14 bit register copies the output of control storage. The low order six bits are used as the control address register and con be loaded from several sources as well as control storage. The output of the high order 8 bits is input to the decode networks and the output of control address register goes to the low order six bits of the execute (XEQ) register.

3 - DECODE NETWORKS

This is a network of gates whose function is to decode the operations and conditions codes. The result of the operations code decoder is to select bits of the XEQ register corresponding to those control lines that are to be raised if the operation is to be executed. The condition code decoder selects which condition is to be tested and enables the XEQ register if it is true. If the condition is false, the XEQ register does not accept the input from the operations code decoder and, in effect, the operation is not executed.

4 - EXECUTE REGISTER (XEQR)

This is a register in which each bit controls the state of a separate control line to the many registers and networks within the machine. The low order six bits are a copy of the control address register one cycle before. The output of these six

bits goes to the I-buss which can be copied into several registers when an operation has a field of immediate data.

5 - MEMORY ADDRESS REGISTER (MADR)

This is a four bit register loaded from bits 6-9 of the META 4 output register associated with control. This register is loaded either at the start of execution or when a LDMADR operation is executed. This register selects the location used in the scratch pad memories of each processor.

6 - SOURCE, ENABLE REGISTER (SENR)

This is a six bit register loaded from the I-buss. The four enable bits select which processors are enabled and thus allowed to change. The two source bits are OR'ed with bits 4 and 5 of the META 4 output register and the result determines what input to the ALU's the data selectors select.

7 - ALU FUNCTION REGISTER (ALUR)

This six bit register is loaded from the I-buss and determines the functions performed by the four ALU's. The first three bits determine the function, the next determines the carry in to each ALU, the next enables the rounding logic, and the last bit inverts the first four bits as they apply to the odd numbered processors. When set the last bit allows addition in the even processors and subtraction in the odd processors. When not set, all ALU's perform the same function.

The functions performed are:

000 F = S
001 F = B
010 F = B minus S minus 1
011 F = $\overline{B}$ + S (OR)
100 F = B•$\overline{S}$ (AND)
101 F = B plus S
110 F = $\overline{B}$
111 F = $\overline{S}$

8 - COUNT REGISTER

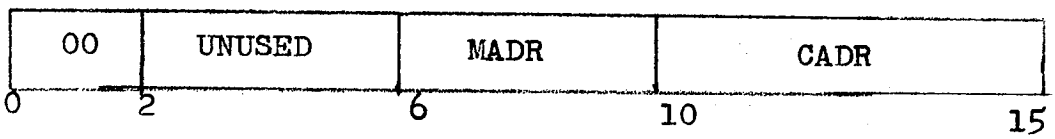
This is a four bit counter loaded from the I-buss. This down counter is tested for zero every time the zero count condition is specified in an operation, and it is then decremented.

9 - NEXT ADDRESS REGISTER

This six bit register adds one to what is on the I-buss and loads the result every cycle. As a result this register always contains the address of the next sequential operation after the operation currently in the XEQ register. When an operation with immediate data is executed, this register is copied into the CADR on the next cycle and the XEQ is disabled until the correct next operation reaches the XEQR. This is why some operations take three cycles.

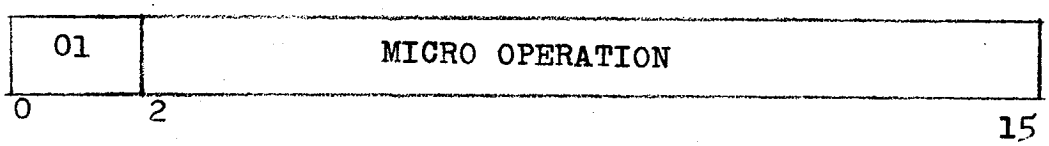
INSTRUCTIONS EXECUTED BY SIMAL E

LOAD MADR AND CADR



This instruction loads the memory address register and the control address register. The contents of control storage address specified by CADR is then loaded into the CSBR.

LOAD AND INCREMENT



This instruction loads the micro operation in bits 2-15 into the control storage addressed by the CADR, then increments the CADR.

BEGIN EXECUTION

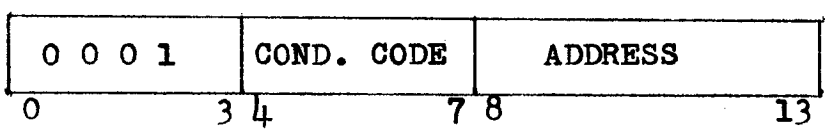
This instruction loads the MADR and CADR registers then begins execution of the micro operation at the location specified in the CADR. The three flag bits are tested by the micro program. See SENR register for explanation of source bits.

MICRO OPERATIONS EXECUTED BY SIMAL E

STOP CC,ADDRESS

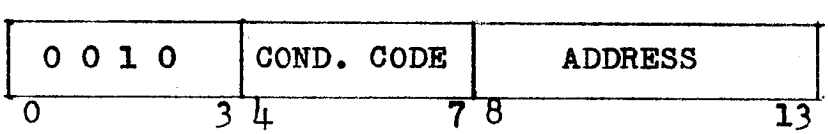
STOP EXECUTION: Stops execution of micro-operations on specified conditon true; else, branches to ADDRESS . Stopping execution causes termination of META 4 Input-Ready and Output-Resume signals; unlocking the input/output registers.

LDB CC,ADDRESS



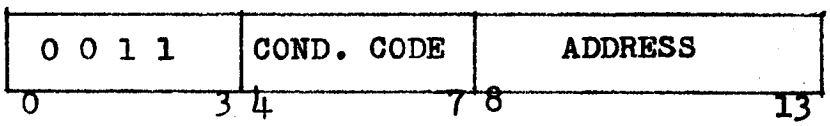
LOAD B REGISTERS: Loads all B registers from their respective F-busses on specified condition true, then branches to ADDRESS.

SRB CC,ADDRESS



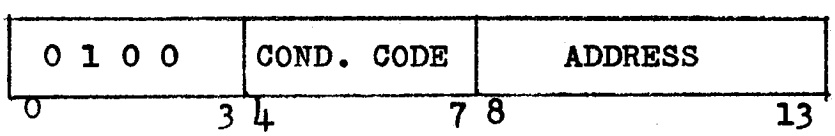
SHIFT RIGHT B REGISTERS: Shifts right algebraeically all B registers on specified condition true, then branches to ADDRESS.

SLB CC,ADDRESS



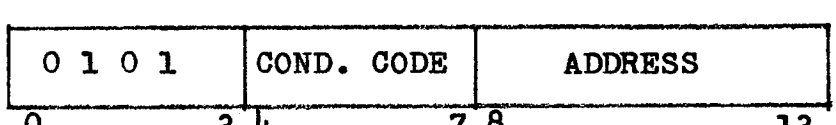
SHIFT LEFT B REGISTERS:Shifts left logically all B registers on specified condition true, then branches to ADDRESS.

LD&STB CC,ADDRESS



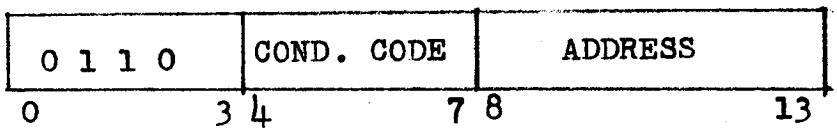
LOAD AND SET B REGISTERS:Loads even B registers from their respective F-busses and sets odd B registers' spill bits 0 on specified condition true. Setting spill bits 0 allows a one to be shifted into the odd B registers during divison. The specified condition assumes two values; one affects operation on B pairs (0,1), and one affects operation on B pairs (2,3). If the specified condition must apply to all B registers, then the two values are the same. A branch to ADDRESS is made.

LDA CC,ADDRESS



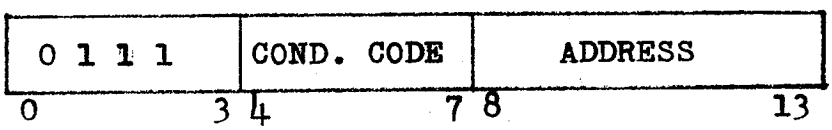
LOAD A REGISTERS:Loads all A registers from their respective F-busses on specified condition true, then branches to ADDRESS.

WRMEM CC,ADDRESS



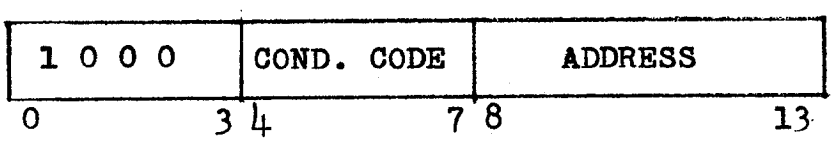
WRITE INTO MEMORY:Writes into scratch pad memories what is on their respective F-busses, then branches to ADDRESS. The address written into is in the Memory Address Register.

SRMOP CC,ADDRESS



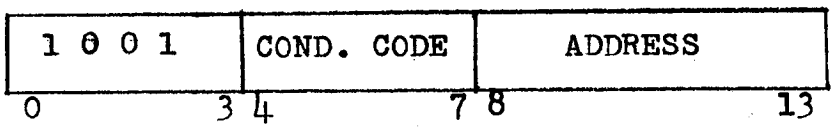
SHIFT RIGHT MOP REGISTER:Shifts right MOP register with specified condition value as input from left, then branches to ADDRESS.

SLMOP CC,ADDRESS



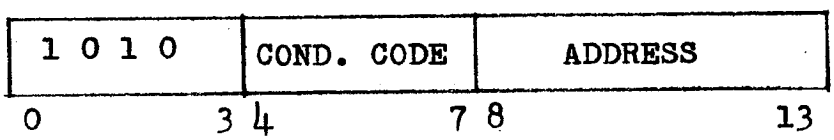
SHIFT LEFT MOP REGISTER:Shifts left MOP register with specified condition value as input from right, then branches to ADDRESS.

LDMADR CC,ADDRESS



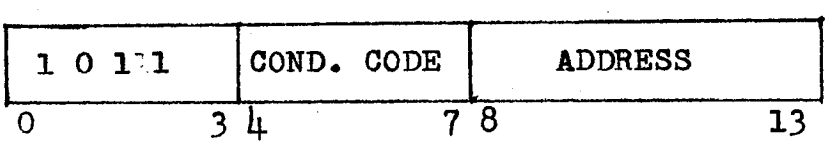
LOAD MEMORY ADDRESS REGISTER:Loads memory address register from META 4 output register bits (6-9) on specified condition true, then branches to ADDRESS.

LDMOP CC,ADDRESS



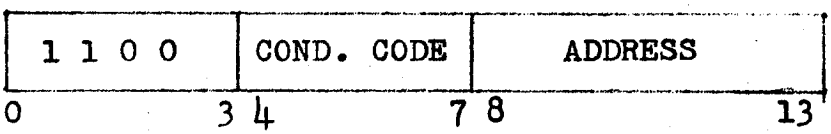
LOAD MOP REGISTER:Loads MOP register from META 4 output register bits (0-15) on specified condition true, then branches to ADDRESS.

RSTB CC,ADDRESS



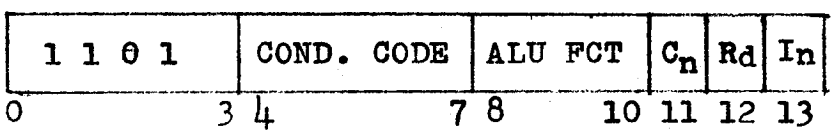
RESET B REGISTERS:Resets B registers on specified condition true, then branches to ADDRESS.

BR CC,ADDRESS



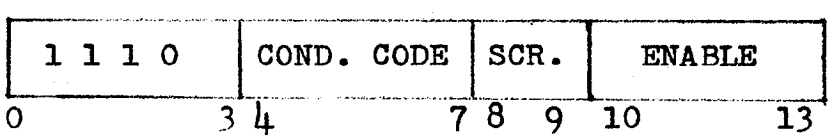
BRANCH:Branches to ADDRESS on specified condition false; else, waits two cycles and continues execution of next sequential operation.

LDALUR CC,FCT,C,R,I



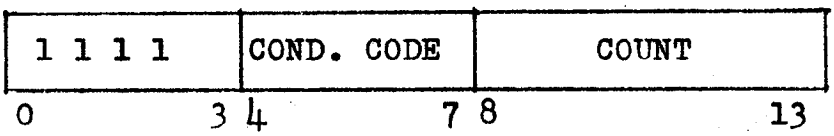
LOAD ALU FUNCTION REGISTER:Loads ALU function register from immediate fields on specified condition true, then waits two cycles and continues execution of next sequential operation. See documentation on this register for further explanation.

LDSENR CC,SC,EN



LOAD SOURCE AND ENABLE REGISTER:Loads source and enable register from immediate field on specified condition true, then waits two cycles and continues execution of next sequential operation. See documentation on this register for further explanation.

LDCTR CC,COUNT



LOAD COUNT REGISTER:Loads count register on specified condition true, then waits two cycles and continues execution af next sequential operation. See documentation on this register for further explanation.

CONDITION CODES

CODE	MNEMONIC	EXPLANATION
0000	NOP	Always equals zero; no operation.
0001	UNCOP	Always equals one; unconditional operation.
0010	DSIGN	Two values: Different sign A&B(0) regs. Different sign A&B(2) regs.
0011	GOES	Two values: Carry Out $\oplus$ A sign(0) Carry Out $\oplus$ A sign(2)
0100	LSIG	Left Significance in any one of the B registers.
0101	ZB	All B registers equal 0 or -1.
0110	REJ	Reject line segment-outside window.
0111	TREJ	Trivial reject line- initally outside window.
1000	MOP 0	Left most bit in MOP.
1001	MOP 15	Right most bit in MOP.
1010	CT	Count equal zero. Count decremented on next cycle.
1011	FLAG1	Bit 1 of I Reg. from META 4.
1100	FLAG2	Bit 2 of I Reg. from META 4.
1101	FLAG3	Bit 3 of I Reg. from META 4.
1110		
1111		