Chapter 14

Instruction logic of the MIDAC[1]

John W. Carr III

The MIDAC, Michigan Digital Automatic Computer [Carr, 1956], was constructed on the basis of the design of the SEAC at the National Bureau of Standards. Its instruction code is particularly of interest because it incorporates the index register concept into a three-address binary instruction. Numbers in this machine are (44, 0, 0)[2] fixed points. The word length is 45 binary digits with serial operation.

Word structure

The data or address positions of an instruction are labeled the α, β, and γ positions. Each contains twelve binary digits represented externally as three hexadecimal digits. Four binary digits, or one hexadecimal digit, are used to convey the instruction modification or relative addressing information. The next four binary digits or single hexadecimal digit represents the operation portion of the instruction. The final binary digit is the halt or breakpoint indicator for use with the instruction.

For example, the 45-binary-digit word

000001100100000011001000000100101100000001011

considered as an instruction would be interpreted as

α	β	γ	abcd	Op	halt
000001100100	000011001000	000100101100	0000	0101	1

In external hexadecimal form this would be written

064 0c8 12c 0 5 –

The above binary word is the equivalent machine representation of the following instruction: "Take the contents of hexadecimal address 064, add to it the contents of hexadecimal address 0c8, and store the result in hexadecimal address 12c. There is *no* modification of the 12-binary-digit address locations given by the instruction. Upon completion of the operation, stop the machine if the proper external switches are energized." The binary combination represented by 5 is the operation code for addition.

Data or addresses

The addresses given by the twelve binary digits in each of the three locations designate in the machine the individual acoustic storage cells and blocks of eight magnetic drum storage cells. The addresses from 0 to 1023 (decimal) or 000 to 3FF (hexadecimal) correspond to acoustic storage cells. The addresses from 1024 to 4095 (decimal) or 400 to FFF (hexadecimal) correspond to magnetic drum storage blocks. In certain operations, however, the addresses 0 to 15 (decimal) or 0 to F (hexadecimal) represent input-output stations rather than storage locations.

These twelve-binary-digit groups will in some cases be modified by the machine in order to yield a final twelve-binary-digit address. The method of processing will depend on the values of the instruction modification digits. After modification, the final result will then be interpreted by the control unit as a machine address.

In some instructions, namely those that perform change of control operations, which involve cycling and counting rather than simple arithmetic operations on numbers, the α and β positions in an instruction are not considered as addresses. In those cases, they are used instead as counters or tallies. In other instructions, which do not require three addresses, but only one or two, the β position is not considered as an address. In these cases, the oddness or evenness of the β address is used to differentiate between two operations having the same operation code digits. That is, the parity of binary digit P22 is used as an extra function designator.

Instruction modification digits

The four binary digits P9-P6 are used as instruction modification or relative addressing digits. Their normal function is relatively simple; nevertheless, the possible exceptions to the general rule can make their behavior complicated. These four digits are labeled

[1] In E. M. Grabbe, S. Ramo, and D. E. Wooldridge (eds.), "Handbook of Automation, Computation, and Control," vol. 2, chap. 2, pp. 115–121, John Wiley & Sons, Inc., New York, 1959.

[2] Carr's triplet notation for: fractional significant digits, digits in exponent, and digits to left of radix point.

the a, b, c, and d digits. Ordinarily the a digit is associated with the α position, the b digit with the β position, and the c digit with the γ position in an instruction.

When binary digit P22 (or the β position) is used in an instruction to represent extra operation information, the instruction modification digit b is ignored. In the case of input and output instructions, when the various address positions represent machine address locations on the drum, input-output stations, or block lengths, and modification of these addresses is not desired in any case, the corresponding relative addressing digits are ignored.

The purpose of the instruction modification digits is to tell the machine whether or not to modify the twelve binary digits making up the corresponding address position in an instruction by addition of the contents of one or the other of two counters. In the normal case, if the a, b, or c digit is a zero, the twelve binary digits in the corresponding position are interpreted, unchanged, as the binary representation of the machine address of the number word to be processed by the instruction.

If one or more of the a, b, or c digits is a one, the contents of one of two auxiliary address counters is added to the corresponding twelve binary digits to yield a final address usually different from that given by the original twelve-digit portion of the instruction word. The addresses are then said to be relative to the counter.

The two counters involved in the address modification feature of the MIDAC are known as the instruction counter and the base counter. In the normal case, if the fourth instruction modification or d digit is a zero, the contents of the instruction counter will be added to the contents of the various twelve-digit addresses (dependent on the values of the a, b, and c digits) before further processing of the instruction. If the a digit is one and the d digit zero, the contents of the instruction counter will be added to the α address; similarly for b and d digits and β address, etc.

If the d digit is a one, the contents of the base counter will be normally added to the contents of the twelve digits in the α, β, and γ positions (again dependent on the values of the a, b, and c digits), before further processing of the results. If the a digit is one and the d digit one, the contents of the base counter will be added to the α address, etc.

The effect of the instruction modification digits may be summarized as follows:

The contents of the two counters will be designated by C_d (d = 0, 1).

C_0 = contents of the instruction counter
C_1 = contents of the base counter

Then the modified addresses α', β', and γ' are related to the α, β, and γ addresses appearing in the instruction by the following:

$$\alpha' = \alpha + aC_d \qquad \beta' = \beta + bC_d \qquad \gamma' = \gamma + cC_d \qquad (a, b, c, d = 0, 1)$$

In certain instructions addresses relative to one of the two counters may be prohibited. Thus, if in a particular instruction α may be relative only to the instruction counter, then for that instruction

$$\alpha' = \alpha + aC_0$$

no matter whether the d digit is a 0 or a 1.

The notation (α'), (β'), or (γ') is used to indicate the word stored in the location whose address is α', β', or γ'.

Instruction counter

The instruction counter is a twelve-binary digit (modulo 4096) counter which contains the binary representation of the address of the instruction which the control unit is processing or is about to process. In normal operation when no change of control operation is being processed, the contents of the instruction counter is increased by one at the completion of each instruction. Thus, normally the next instruction to be processed is stored in the acoustic storage cell immediately following the cell which contains the present instruction.

A change of control operation is one which selects a next instruction not stored in sequence in the acoustic storage. That is, at the completion of such instructions the contents of the instruction counter is not increased by one, but instead is changed entirely.

Base counter

The base counter is a second twelve-binary-digit counter (modulo 4096), physically identical to the instruction counter, which contains the binary representation of a base number or tally. Unlike the instruction counter, however, the base counter does not sequence automatically, but remains unchanged until a change of base instruction is processed. This counter serves two primary purposes, dependent on the usage to which it is put:

1 It may contain the address of the initial word in a group, thus serving as a base address to which integers representing the relative position of a given word in the group of words may be added by using the address modification digits.

2 It may contain a counter or tally which can be increased by a base instruction. This instruction makes use of the address modification digits to change the counter so as to count the number of traversals of a particular cycle of instructions.

Instruction types

Instructions used in MIDAC can be divided into three categories: change of information, change of control, and transfer of information. The first category can be further subdivided into arithmetic and logical instructions. In the arithmetic instructions are included addition, subtraction, division, various forms of multiplication; power extraction, number shifting; and number conversion instructions. The sole logical instruction is extract, which modifies information in a nonarithmetic fashion.

The transfer of information or data transfer instructions include transfers of individual words or blocks of words into and out of the acoustic storage and drum and magnetic tape control.

The possible change of control instructions includes two comparisons that provide different future sequences dependent on the differences of two numbers. In the compare numbers or algebraic comparison, the difference is an algebraic, signed one. In the compare magnitudes or absolute comparison, the difference is one between absolute values. Two other instructions, file and base, perform other tasks beside transferring control. The file instruction transfers control unconditionally. The file instruction files or stores the contents of the base or instruction counter in a specific address position of a particular word in the storage. The base or tally instruction provides a method for referring addresses automatically relative to the address given by the base counter, irrespective of its contents. The base instruction also gives a conditional transfer of control.

The nineteen MIDAC instructions can be described functionally as follows:

Change of information

1 **Add.** $(\alpha') + (\beta')$ is placed in γ'. Result must be less than 1 in absolute value.

2 **Subtract.** $(\alpha') - (\beta')$ is placed in γ'. Result must be less than 1 in absolute value.

3 **Multiply, Low Order.** The least significant 44 binary digits of $(\alpha') \times (\beta')$ are placed in γ'.

4 **Multiply, High Order.** The most significant 44 binary digits of $(\alpha') \times (\beta')$ are placed in γ'.

5 **Multiply, Rounded.** The most significant 44 binary digits of $(\alpha') \times (\beta') \pm 1 \cdot 2^{-45}$ are placed in γ'. The $1 \cdot 2^{-45}$ is added if $(\alpha') \times (\beta')$ is positive, and subtracted if $(\alpha') \times (\beta')$ is negative.

6 **Divide.** The most significant 44 binary digits of $(\beta')/(\alpha')$ are placed in γ'. (Note the inversion of order of α and β.) Result must be less than 1 in absolute value.

7 **Power Extract.** The number $n \cdot 2^{-44}$ is placed in γ' where n is the number of binary 0's to the left of the most significant binary 1 in (α'). The b digit is ignored; β may be any even number. If (α') is all zeros, zero is placed in γ'.

8 **Shift Number.** The 44 binary digits immediately to the right of the radix point in $(\alpha') \cdot 2^{(\beta') \cdot 2^{44}}$ are placed in γ'. The result, in γ', is the equivalent of shifting (α') n places, where $n \cdot 2^{-44} = (\beta')$ and n positive indicates a shift left, n negative a shift right. If $|n| \geqq 44$, zero is placed in γ'.

9 **Extract or Logical Transfer.** Those binary digits in (γ'), including the sign digit, whose positions correspond to 1's in (β') are replaced by the digits in the corresponding positions of (α').

10 **Decimal to Binary Conversion.** This operation may be interpreted in two ways: (*a*) (α') is considered as a binary-coded-decimal integer times 2^{-44}. It is converted to the equivalent binary integer times 2^{-37} and the result is placed in γ', or (*b*) (α') is considered as a binary-coded-decimal fraction, D. It is converted into an intermediate binary fraction, B_i, such that $B_i = D \times 10^{11} \times 2^{-37}$ and the result placed in γ'. To obtain B, the true binary equivalent of D, B_i must be multiplied by $(10^{-11} \times 2^{37})$. However, since this factor is greater than 1 and therefore cannot be represented in the machine, two operations must be performed. For example,

$$B_i \times (10^{-11} \times 2^{37} - 1) = B_j$$
$$B = B_i + B_j$$

Here the b digit is ignored, and β may be any *even* number.

11 **Binary-to-Decimal Conversion.** (α'), considered as a binary fraction, is converted into the equivalent eleven-digit binary-coded-decimal fraction. The result is placed in γ'. The b digit is ignored, and β may be any odd number.

Change of control

12 **Compare Numbers.** γ can be relative only to the instruction counter. If $(\alpha') \geqq (\beta')$, the contents of the instruction counter are increased by one as is normally done at the end of each instruction. If $(\alpha') < (\beta')$, the contents of the instruction counter are set to γ'.

13 **Compare Magnitudes.** γ can be relative only to the instruction counter. If $|(\alpha')| \geqq |(\beta')|$, the contents of the instruction counter are increased by one as is normally done at the end of each instruction. If $|(\alpha')| < |(\beta')|$, the contents of the instruction counter is set to γ'.

14 **Base or Tally.** The d digit is ignored. α and β may be relative only to the base counter, γ only to the instruction counter. If $\alpha' \geqq \beta'$, the contents of the base counter are set to zero and the contents of the instruction counter increased by one as usual. If $\alpha' < \beta'$, the contents of the base counter are set to α' and the contents of the instruction counter to γ'. (*Note.* The comparisons made here are of addresses themselves, not their contents.)

15 **File.** β may be any odd number. α and γ may be relative only to the instruction counter.

If d = 0, the contents of the instruction counter increased by one is placed in the γ position of (α'), and the instruction counter is set to γ'.

If d = 1, the contents of the base counter is placed in the α position of (α'), and the instruction counter is set to γ'. In addition, if b = 1, the contents of the base counter is set to zero; if b = 0, the contents of the base counter is not changed.

Transfer of information

16 **Read In.** The a digit must be 0; the b digit is ignored. If β is in the range 0 to 7 (decimal) or 000 to 007 (hexadecimal) α words are read into the acoustic storage from input-output station β. The first word read in is placed in γ', the second in $\gamma' + 1$, etc. If β is in the range 1024 to 1791 decimal (400 to 6FF hexadecimal), α words are read into the acoustic storage from the drum starting with the first word in the drum block whose address is β. The first word is placed in γ', the second in $\gamma' + 1$, etc.

17 **Read Out.** The a digit must be 0, the c digit is ignored. Starting with (β'), read out α consecutive words from the acoustic storage to input-output station γ, if γ is in the range 0 to 7 decimal (000 to 007 hexadecimal), or to the drum starting at the beginning of the drum block whose address is γ, if γ is in the range 1024 to 1791 decimal (400 to 6FF hexadecimal).

16 **Alphanumeric Read In.** The a digit must be 1; the b digit is ignored. If β is in the range 0 to 7 (decimal) or 000 to 007 (hexadecimal) α characters are read into the acoustic storage from input-output station β. The first character read in is placed in γ', the second in $\gamma' + 1$, etc. Each character occupies the six most significant digit positions of the register into which it is read; the other positions are set to zero. This operation may not be used to read words from the drum into the acoustic storage.

17 **Alphanumeric Read Out.** The a digit must be 1; the c digit is ignored. Starting with (β'), read out α consecutive characters from the acoustic storage to input-output station γ; γ must be in the range 0 to 7 (decimal) or 000 to 007 (hexadecimal). This operation may not be used to read words from the acoustic storage onto the drum.

18 **Move Tape Forward.** (a, b, c and d digits are ignored.) β may be any *even* number; γ must be in the range 0 to 15 decimal (000 to 00F hexadecimal). The magnetic tape at input-output station γ is moved forward n blocks where

$$n = \left[\frac{\alpha - 1}{8}\right] + 1$$

that is, one plus the integral part of $\alpha - \frac{1}{8}$, or the number of blocks that include α words.

19 **Move Tape Backward.** (a, b, c, and d digits are ignored.) β may be any *odd* number; γ must be in the range 0 to 15 decimal (000 to 00F hexadecimal). The magnetic tape at input-output station γ is moved backward n blocks where

$$n = \left[\frac{\alpha - 1}{8}\right] + 1$$

that is, one plus the integral part of $\alpha - \frac{1}{8}$, or the number of blocks that include α words.

References

CarrJ56. SEAC computer references: AinsE52; AlexS51; ElboR53; GreeS52, 53; HaueR52; PikeJ52; SerrR62; ShupP53; SlutR51. DYSEAC computer references: LeinA54.