Combelic

# INTERPRETED INSTRUCTION CODE OF THE MIT CS II COMPUTER

(See definition of symbols in Table I)

| Instr.## | Decimal | Meaning | Definition | Alarm# |
|---|---|---|---|---|
| | 0 | (undefined instruction) | | E |
| its al+c | 1 | (cycle) transfer N(MRA) into (al+2i,al+2i+1) | N(MRA)$\longrightarrow$N(al+2i) | B |
| iex al+c | 2 | (cycle) exchange | N(MRA)$\longleftrightarrow$N(al+2i) | B |
| ica al+c | 3 | (cycle) clear MRA; add N(al+2i) | N(al+2i)$\longrightarrow$N(MRA) | |
| ics al+c | 4 | (cycle) clear MRA; subtract N(al+2i) | -N(al+2i)$\longrightarrow$N(MRA) | |
| iad al+c | 5 | (cycle) add | N(MRA)+N(al+2i)$\longrightarrow$N(MRA) | D' |
| isu al+c | 6 | (cycle) subtract | N(MRA)-N(al+2i)$\longrightarrow$N(MRA) | D' |
| imr al+c | 7 | (cycle) multiply and roundoff | N(MRA)xN(al+2i)$\longrightarrow$N(MRA) | D',K |
| idv al+c | 8 | (cycle) divide | N(MRA)$\div$N(al+2i)$\longrightarrow$N(MRA) | C',D',E |
| isp al+c | 9 | (cycle) transfer of control | take the next instruction from reg.(al+i) and continue from there | E |
| isc j** | 10 | select counter | select cycle count line j | A |
| icr m** | 11 | cycle reset | set i=+0, n=m | |
| ict al | 12 | cycle count | increase i by 1; if $\|i_k\| \geq \|n\|$ reset i=+0 and take next instruction in sequence; if $\|i_k\| < \|n\|$ take next instruction from register al | E,J |
| iat al | 13 | add and transfer | add C(index reg.) to the C(al) and store the result in index register and register al | I |
| iti al | 14 | transfer index digits | transfer the right 11 digits of the index reg. into the right 11 digits of register al | I |
| sp al | 15 | transfer of control | take the next instr. from reg.al and continue from there in uninterpretive mode(sp0 stops computer) | E |
| ici m** | 16 | cycle increase | increase contents of index register by m | G |
| icd m** | 17 | cycle decrease | decrease contents of index register by m | H |
| icx al | 18 | cycle exchange | exchange C(index reg.) with C(al) and exchange C(criterion register) with C(al+1) | |
| ita al | 19 | transfer address | replace the address section of the instruction in register al with the address that is one more than the address of the register containing the last isp(or icp with N(MRA)neg.) | |
| icp al | 20 | conditionally transfer control (conditional program) | take the next instruction from reg. al and continue from there, if N(MRA) is neg.; if N(MRA) is pos. take next instruction in sequence | E |
| *its al | 21 | transfer N(MRA) into (al,al+1) | N(MRA)$\longrightarrow$N(al) | B |
| *iex al | 22 | exchange N(MRA) with N(al) | N(MRA)$\longleftrightarrow$N(al) | B |
| *ica al | 23 | clear MRA; add N(al) | N(al)$\longrightarrow$N(MRA) | |
| *ics al | 24 | clear MRA; subtract N(al) | -N(al)$\longrightarrow$N(MRA) | |
| *iad al | 25 | add | N(MRA)+N(al)$\longrightarrow$N(MRA) | D,L |
| *isu al | 26 | subtract | N(MRA)-N(al)$\longrightarrow$N(MRA) | D,L |
| *imr al | 27 | multiply and roundoff | N(MRA)xN(al)$\longrightarrow$N(MRA) | D,K |
| *idv al | 28 | divide | N(MRA)$\div$N(al)$\longrightarrow$N(MRA) | C,D,K |
| isp al | 29 | transfer control | take the next instr. from reg. al and continue from there | E |
| | 30-31 | (undefined instructions) | | E |

* The buffer letter "b" may be used with these instructions only.
**m and j are positive integers less than 2,048.
# Consult Alarm Table in this memo.
##For Output Instructions see Table III.

---

Table I - DEFINITION OF SYMBOLS

| Symbol | Meaning |
|---|---|
| MRA | multiple register accumulator |
| al | let al represent any floating address, absolute address or buffer address |
| N(MRA) | the number in the MRA before the instruction is obeyed |
| N(al) | the number stored in registers al and al+1 before the instruction is obeyed |
| C(...) | contents of ... |
| i | C(index register) |
| $i_k$ | new C(index register) |
| n | C(criterion register) |
| N(al+2i) | the number stored in registers al+2i and al+2i+1 before the instruction is obeyed |
| ⟶ | replaces |
| clear... | set the contents of ... to zero |
| Buffer | block of three registers containing numbers in same form as in MRA |

---

Table II - ALARMS (C',D' are same as C, D except that al+c replaces al)

Check Order Alarms

(A) Counter not provided for by the PA is selected (this can occur only if the "j" in isc j has been modified by the program).

(B) Exponent of $N(MRA) \geq 2^j$ where j refers to the (30-j,j) notation (provided al is not a buffer). See * above.

(C) C(al) = 0

(D) $0 < |C(al)| < \frac{1}{2}$

(E) When control is transferred to an undefined instruction, an alarm occurs on the undefined instruction.

Arithmetic Overflow Alarms

(G) C(index register) + m > 32,767

(H) C(index register) - m < -32,767

(I) |C(index register) + C(al)| > 32,767

(J) |i| = 32,767 before the ict is executed

(K) $|Result| > 7.0 \times 10^{9363}$ or $|Result| < 7.1 \times 10^{-9364}$

(L) If al is a buffer, then alarm K could occur

---

Table III - OUTPUT INSTRUCTIONS

A. Specifications using either iTOA, iMOA or iSOA

| | | |
|---|---|---|
| iTOA abcdefg | Record N(MRA) on direct printer | See below for description of a, b, c, d, e, f, g |
| iMOA abcdefg | Record N(MRA) on delayed printer | |
| iSOA abcdefg | Record N(MRA) on scope(film) | |

Ex: iTOA + i 12.345 x $2^{-3}$ x $10^{2}$ c
(a)(b)(c)(d)(e) (f) (g)

(a) Sign — Meaning

| | |
|---|---|
| + | all numbers will be preceded by sign |
| - | only negative numbers will be preceded by sign |
| nothing | numbers will not be preceded by sign |

(by "nothing" we do not mean that the word "nothing" is written out)

(b) Initial Zeros — Meaning

| | |
|---|---|
| i | initial zeros will be skipped (except for the one just to the left of the decimal point) and the first significant digit of the number will be printed at the extreme left of the column. (Thus the left-most digit of all numbers in a column will be lined up.) |
| p | initial zeros will be replaced by spaces. (Thus numbers in a column are arranged so that the decimal points fall in a line.) |
| n | all numbers will be printed in a normalized form, i.e., all numbers are multiplied by a power of 10 such that the first non-zero significant digit always falls in the same relative position with respect to the decimal point. The number printed is followed by a vertical bar followed by the signed power of 10 that the number is to be multiplied by. |
| nothing | initial zeros will be printed as zeros. |

(c) Digits Left

The programmer indicates the number of digits he wishes to have printed to the left of the decimal point by actually writing a sample number containing the same number of digits to the left of the decimal point as he wishes printed. Thus

iTOA + 123.4567

specifies that the programmer wishes to have his numbers printed with 3 digits to the left of the decimal point. (The magnitude of the digits one writes in the sample number has no effect whatsoever on the program.) If the number actually contains more than three digits to the left of the decimal point, all these digits will automatically be printed out.

(d) Decimal Point — Meaning

| | |
|---|---|
| . | A decimal point will appear in all numbers |
| nothing | No decimal point will be printed. (Used when programmer desires only integral part of number.) |
| r | No decimal point will be printed. (Used when programmer expects all results to be less than one; if the number in the MRA should unexpectedly exceed unity, then the integral part of the number will also be printed out but no decimal point will separate the integral from the fractional part of the number. |

(e) Digits Right

If a programmer were to use the sample number illustrated in (c), he would get four digits printed out to the right of the decimal point (or to the right of where the decimal point should be).

(f) Scale Factors

Powers of 2 and 10 may be used as a scale factor which will be applied to a number before the number is printed out:

(1) Every factor must be preceded by a lower case x.

(2) $|\alpha|, |\beta| \leq 99$

(g) Terminal Characters (character used to terminate a number)

| | Meaning |
|---|---|
| s | space |
| ss | 2 spaces |
| sss | 3 spaces |
| ssss | 4 spaces |
| c | carriage return |
| t | tab |
| nothing | carriage of typewriter will remain exactly where it was after the last number was typed. |
| f | format (see section B) |

EXAMPLES

| | N(MRA) | OUTPUT REQUEST | PRINTED RESULT |
|---|---|---|---|
| (1) | -7.953261 | iTOA + 123.1234s | -007.9532 space |
| | | iTOA + i123.1234 | -7.9532 |
| | | iTOA p123.1234c | 7.9532 car. ret. |
| | | iTOA - n123.1234t | -795.3261\|-02 tab |
| (2) | +795.3261 | iTOA 123.123ss | 795.326 space space |
| | | iTOA-i1234.5x10$^2$c | 79532.6 car. ret. |
| | | iTOA+p12r34 | +79532 |
| | | iTOA n1.234t | 7.953\|+02 tab |

B. Format specification (By using this facility, the programmer avails himself of an automatic device for obtaining a suitable layout of his output data.)

If "f" is used as a terminal character in an output request (see Table III-section (A)-g) then the instruction and 3 program parameters iFOR $\alpha\beta\gamma$ must precede the output instruction containing the "f". (This will furnish the CS output section with the necessary layout information before a number is printed out.)

$\alpha$ represents the number of words/line (maximum number of characters per line is 155)

$\beta$ represents the number of spaces between words (maximum is 5) (A tab is obtained by setting $\beta \geq 5$)

$\gamma$ represents the number of words per block (The maximum $\gamma$ is 32,767. Since the block counter is automatically reset after each block is completed, the upper limit, 32,767, for $\gamma$ is not a significant limitation.)

ex. 1 Supposing the programmer wishes to have 2500 words typed out and uses iFOR with the output request iTOA+12.345f. This will give 12 words per
+12
+2
+400
line, 2 spaces between words and 400 words per block. The blocks are separated by 2 carriage returns. In this example there will be six blocks of 400 words and one block of 100 words. (The programmer should provide carriage returns at the end of his print-out

if that doesn't coincide with the end of a block. This carriage return order is described in the Special Characters Section of E-516-2.)

---

Table IV - AUXILIARY EQUIPMENT INSTRUCTIONS

Magnetic Drum

The drum may be used with the instructions iDIB, iDOB where D designates Drum, I designates Input (drum to MCM), O designates Output (MCM to drum), and B designates Binary. Each of these forms must be followed by 3 program parameters as described below and in the order listed:

(1) Initial address of Magnetic Core Memory (MCM)
(2) Initial address of Magnetic Drum Storage (MDS)
(Drum registers available are 2,048-20,479)
(3) Length of block being transferred to or from MCM.

Example:

```
iDIB    [This will transfer the contents of registers 12280-12321 of MDS to registers
+900     900-941 of MCM (interpreted return control). If registers 900-941 are
+12280  double-length numbers, then 21 of them will be transferred.]
+42
```

TABLE V - OPERATING TIME, IN $\mu$ SEC., OF CS II INSTRUCTIONS

| | Neither "C" nor "b" blocks used | "C" block used | "b" block used | "b" & "C" blocks used | Neither "C" nor "b" blocks used minimum* |
|---|---|---|---|---|---|
| ica | 654 | 694 | 718 | 758 | |
| ics | 678 | 718 | 742 | 782 | |
| iad | 2007# | 2047# | 2089# | 2129# | 850 |
| isu | 2030# | 2070# | 2110# | 2150# | 880 |
| imr | 1441# | 1481# | 1521# | 1561# | |
| idv | 2203# | 2227# | 2267# | 2307# | |
| isp | 465 | 505 | 529 | 569 | |
| iex | 1275 | 1315 | 1339 | 1379 | |
| its | 901 | 941 | 965 | 1005 | |
| icp | $^{+}_{281}$ $^{-}_{472}$ | $^{+}_{321}$ $^{-}_{512}$ | $^{+}_{345}$ $^{-}_{572}$ | $^{+}_{385}$ $^{-}_{612}$ | |
| ita | 360 | 384 | 408 | 448 | |
| icx | x** | 554 | x | 618 | |
| icd | x | 384 | x | 448 | |
| ici | x | 361 | x | 425 | |
| iti | x | 385 | x | 449 | |
| iat | x | 440 | x | 504 | |
| icr | x | 360 | x | 424 | |
| isp c | x | 673 | x | 673 | |
| idv c | x | 2331# | x | 2331# | |
| imr c | x | 1580# | x | 1580# | |
| isu c | x | 2183# | x | 2183# | 1016 |
| iad c | x | 2159# | x | 2159# | 990 |
| ics c | x | 815 | x | 815 | |
| ica c | x | 791 | x | 791 | |
| iex c | x | 1411 | x | 1411 | |
| its c | x | 1045 | x | 1045 | |
| IN-OUT | 361 | 401 | 425 | 465 | |
| IN-SPX | 361 | 401 | 425 | 465 | |
| ict | x | 434 | x | 498 | |
| isc | x | 851 | x | 915 | |
| ica b | x | x | 974 | 1014 | |

TABLE V - (Cont.)

| | Neither "C" nor "b" blocks used | "C" block used | "b" block used | "b" & "C" blocks used | Neither "C" nor "b" blocks used minimum* |
|---|---|---|---|---|---|
| ics b | x | x | 1017 | 1057 | |
| its b | x | x | 983 | 1023 | |
| iex b | x | x | 1001 | 1041 | |
| iad b | x | x | 2068# | 2108# | |
| isu b | x | x | 2184# | 2224# | |
| imr b | x | x | 1716# | 1756# | |
| idv b | x | x | 2464# | 2504# | |

\# average for operating on positive and negative numbers

* in addition the addend and augend, and in subtraction the minuend and subtrahend do not affect each other because of the great disparity in magnitude (minimum col.)

** x indicates that the instruction doesn't have any meaning in that column