RIP-3000
RECOMP III INTERPRETIVE PROGRAM

by

G. Howell

RIP-3000

REISSUED: 28 JULY 1965

AUTONETICS
A DIVISION OF NORTH AMERICAN AVIATION, INC.

CONTENTS

ILLUSTRATIONS

GLOSSARY

ACCUMULATOR

An internal storage device which holds one of the operands prior to executing an arithmetic instruction and which usually retains the result of the instruction.

ADDRESS

A decimal number in the range. 0 through 3000, which identifies a particular location in the RIP-3000 memory.

DUMP

To output the contents of a block of memory on punched paper tape or on the typewriter.

INDEX REGISTER

An internal storage device which holds a value by which an instruction address may be modified prior to execution.

LOOP

A sequence of instructions which is executed repetitively for a specified number of times before proceeding with the next part of the program.

MEMORY

The main internal storage area in a computer, used to store both instructions and data.

OVERFLOW

The generation of a value beyond the capacity of the accumulator through an arithmetic operation.

PROGRAM

A sequence of instructions to do a particular problem.

SUBROUTINE

A sequence of instructions which performs some well-defined function and which is used in common by more than one program.

TRACE

A mode of program operation in which each instruction, as it is executed, is printed out along with sufficient information to define the effect of the instruction.

TRANSFER

A programmed departure from the linear sequence in which instructions are stored in memory.

WORD

The contents of one internal storage location. In RIP-3000 programs, a word may contain one instruction, one data value, or five alphanumeric characters.

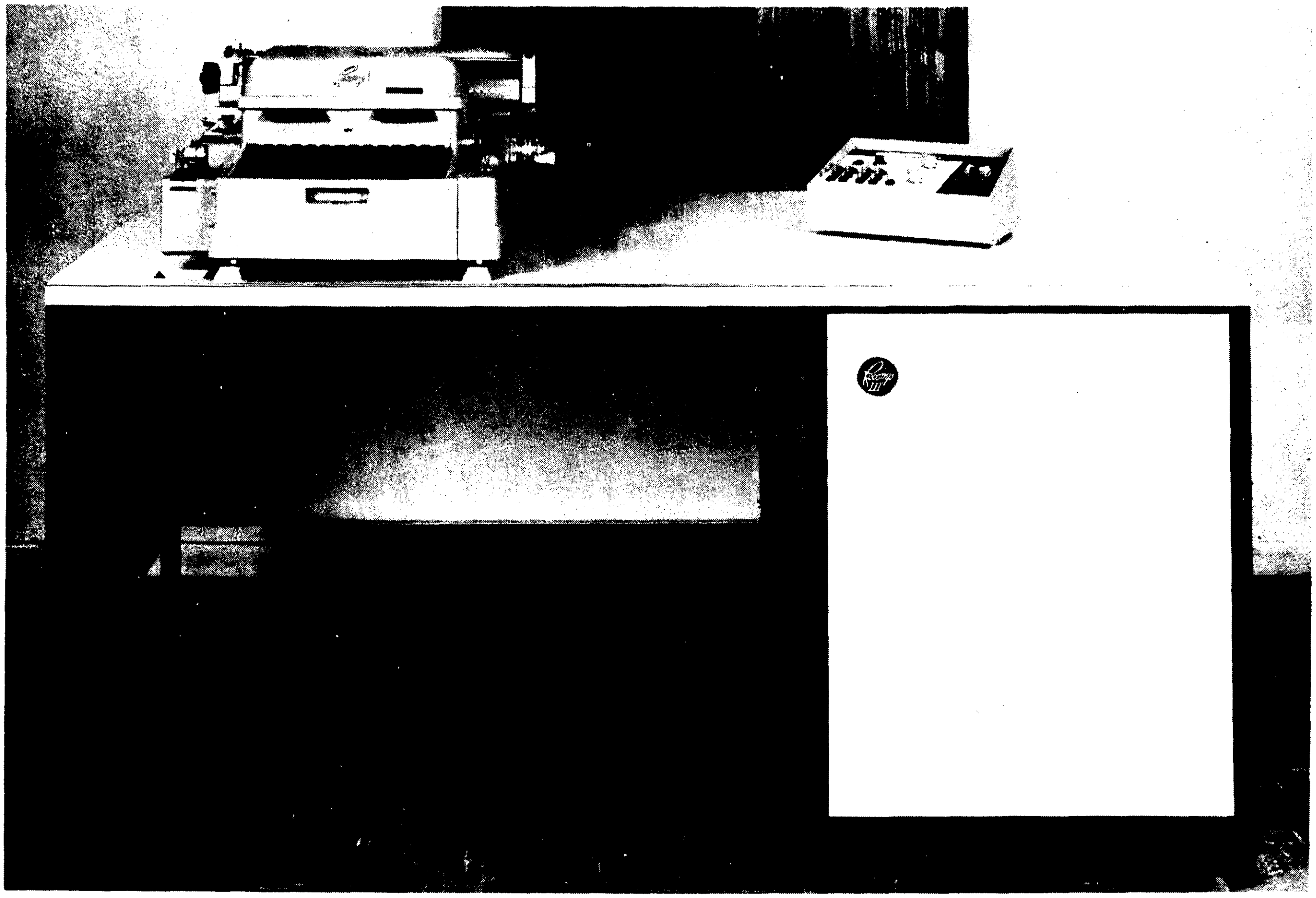


Figure 1.

I. INTRODUCTION

The RIP-3000 Interpretive Program for the Autonetics RECOMP III computer provides the facility for programming in a simplified and abbreviated coding language. Little knowledge of the internal functions of the computer is required. A RIP-3000 language program consists of a list of instructions to the RIP-3000 Interpreter defining the operations to be performed. The instructions are written in the sequence in which they are to be executed. Each required operation consists of a single, meaningful, alphabetic character or special symbol. Addresses and numerical data are expressed in decimal form. The programmer need not be familiar with any number system other than decimal.

RIP-3000 interprets each instruction and provides the necessary machine language instruction(s) to perform the required operation. In many cases, a single instruction is sufficient to direct RIP-3000 to perform a complex mathematical function. The programmer has at his disposal nine index registers which are maintained by RIP-3000 for address modification and loop control.

The RIP-3000 Interpretive Program is a powerful and versatile tool which enables the solution of complex problems with a minimum of coding time and effort. It is particularly valuable where the computer is used on an open shop basis or where programming experience is limited.

II. THE RIP-3000 INTERPRETIVE PROGRAM

GENERAL

Programming in the RIP-3000 language is simple and straightforward without sacrificing flexibility of approach in the solution of a problem. A full 3000 words of memory are available for program and data storage. The remainder of memory is occupied by the Load/Start routine and by RIP-3000 and its subroutine library.

A program to be processed by RIP-3000 may be entered via either the Flexowriter Keyboard or the tape reader. Each RIP instruction word, as it is entered, is converted to a transfer to an appropriate sequence of instructions within the Interpreter. The converted instruction is then stored in its assigned memory location relative to the first word of the program. When the entire program has been processed in this manner, it is in memory, ready for execution.

Once this input phase of RIP processing is complete, the programmer may elect to dump the processed program on tape. Future runs of the program may then be made without undergoing this initial input processing phase. The programmer also has the option of listing the program symbolically on the typewriter.

WORD FORMATS

A RIP-3000 instruction word contains from one to three parts, depending upon the operation specified. The first part of every instruction is a one-character operation code. For most instructions, this is followed by an address specifying the memory location of an operand. In some cases, this second part contains the operand or an identification key rather than an address. The third part is a one-digit index tag and is used only when an index register is involved in the execution of the instruction.

The index tag must not be used with operations t, w, or m. Following are some examples of typical instruction words:

+ 2000   Add the contents of location 2000 to the contents of the accumulator.

l 75 2   Load the value, 75, into index register 2.

p   Make the sign of the accumulator positive.

Data words and constants are input and output in decimal form. On input, a value is expressed as a signed decimal number, including a decimal point where appropriate, and may include a power of 10 by which the number is to be multiplied. All output data is expressed as sign, one-digit integer, decimal point, eight-digit fraction and signed power of 10 by which the number is to be multiplied.

Following are some typical data words:

3724	Integer. Plus sign may be omitted. -- Input format only.
-37.24	Integer and fraction -- Input format only.
-23.845+3	Integer, fraction, and power of 10 -- Input format only.
+2.53972648 +04	Integer, fraction, and power of 10 -- Output format.

OPERATION CODES

There are 31 operation codes available in the RIP-3000 repertoire. Each operation code is expressed as a single symbol. The symbols have been chosen for their mnemonic significance. Thus, the symbol "+" stands for "add"; the letter "s" stands for "store"; etc. The operation code directs RIP-3000 to set up the machine instructions necessary to perform some defined function.

The operation code is the first character entered for each RIP-3000 instruction. Any symbol in this position which is not a part of the RIP-3000 repertoire will cause "bad" to be printed on the typewriter. RIP-3000 will now accept the corrected symbol.

An operation code which does not require an address (z, p, n, e, d, or q) may be followed immediately by a carriage return or a comma, either of which defines the end of the instruction word.

ADDRESSES

The address portion of an instruction is entered after the operation code. It may be preceded by any number of spaces, if desired. Addresses are entered in decimal, and must be in the range 0 through 3000. The address, 3000, refers to the RIP accumulator.

An address which is to be followed by an index tag must be terminated by one or more spaces. Otherwise, it is terminated by a comma or carriage return, either of which defines the end of the instruction.

An erroneous address may be corrected before termination by following it with a slash (/) or bracket (]), followed by the correct address. Any address outside the range 0 through 3000 will cause "bad" to be printed on the typewriter. The instruction must now be re-entered from the beginning.

Any non-numeric characters, except space, carriage return, comma, bracket or slash, will be ignored if included as part of an address.

INDEXING

RIP-3000 provides nine index registers for the purpose of loop control, counting, and sequential list processing. Indexing is applied to an instruction by entering the index tag (1 through 9), following the space(s) which defines the end of the address. When an instruction is indexed, the effective address is determined, each time the instruction is executed, by subtracting the current contents of the specified index register from the original instruction address.

For example, if index register 3 contains 20, the instruction, + 2300 3, would add the contents of location 2280 to the contents of the accumulator.

CAUTION

Indexing must not be applied to the Transfer (t), Wait and Transfer (w), and Mark and Transfer (m) instructions. Inclusion of the index tag in these cases will completely disrupt the intended program sequencing.

An erroneous index tag may be corrected before termination by following it with a slash (/), followed by the correct tag. Any tag over 9 will cause "bad" to be printed on the typewriter. The instruction must now be re-entered from the beginning. The index tag, if used, is always followed by a carriage return or a comma to signify the end of the instruction.

PROGRAM STORAGE

To process or operate a program with RIP-3000, the Load/ Start routine must be in memory in locations 0000_8 through 0177_8.

RIP-3000 occupies locations 0200_8 through 2067_8. This leaves 3000_{10} locations available to the program being processed. This area is referred to as the RIP memory, and is addressed in decimal as locations 0000_{10} through 2999_{10}. Any location within this area may be specified as the starting location to input a program or a group of instructions or constants. Each such group is stored in memory in the sequence in which it is entered until a new starting location is given. RIP-3000 maintains its own location counter for this storage, separate from the computer's location counter.

If this RIP location counter is stepped beyong 2999, it returns to 0000. Thus, proper program sequencing requires that no instruction except a transfer instruction (t or w) be located in 2999.

MATHEMATICAL FUNCTION SUBROUTINES

Included as a part of RIP-3000 are seven mathematical function subroutines (sine, cosine, log base 10, log base e, 10^X, e^X and arctangent.) A single instruction (g), with an identification key in the address field, may be used to execute any of these subroutines.

Provision has been made for the user to add up to five additional subroutines to the RIP-3000 library. The procedure for incorporating a new subroutine is described in Appendix 1.

OVERFLOW

An arithmetic operation which generates a result beyond the capacity of one computer word results in a condition known as "overflow." The maximum absolute value which may be contained in a word is approximately 10^{38}. The minimum non-zero absolute value which may be contained is approximately 10^{-38}. If this range is exceeded in executing a program, the computer will halt and an error note will be printed giving the location of the instruction which generated the error.

The RIP-3000 operations which can cause overflow are +, -, ., /, v, ↑ , r, and g.

III. THE INSTRUCTION LIST

ARITHMETIC OPERATIONS

Op. Code	Address	Index Tag	Meaning
$\boxed{]}$	0000 through 3000	Yes	Clear and Add

The contents of the accumulator are replaced by the contents of the addressed memory location. The word in memory remains intact. The address may be modified by indexing.

An address of 3000 (RIP accumulator) will cause the instruction to do nothing, unless modified by an index register.

Op. Code	Address	Index Tag	Meaning
$\boxed{+}$	0000 through 3000	Yes	Add to Accumulator

The contents of the addressed memory location are added to the contents of the accumulator, and the sum is left in the accumulator. The word in memory remains intact.

The address may be modified by indexing.

An address of 3000 will cause the instruction to double the contents of the accumulator, leaving the result in the accumulator.

Op. Code	Address	Index Tag	Meaning
$\boxed{-}$	0000 through 3000	Yes	Subtract from Accumulator

The contents of the addressed memory location are subtracted from the contents of the accumulator, and the difference is left in the accumulator. The word in memory remains intact.

An address of 3000 will cause the accumulator contents to be subtracted from the accumulator, leaving zero in the accumulator.

The address may be modified by indexing.

EXAMPLE

Assume the following:

Location 2000 contains the value, A.
Location 2001 contains the value, B.
Location 2002 contains the value, C.

The value, 2(A - B) + C, will be computed by the following sequence of instructions, leaving the result in the accumulator.

Location	Operation	Address	Index	Remarks
1000	]	2000		A to accumulator
1001	-	2001		A-B in accumulator
1002	+	3000		2(A-B) in accumulator
1003	+	2002		2(A-B) + C in accumulator

Op. Code	Address	Index Tag	Meaning
$\boxed{\cdot}$	0000 through 3000	Yes	Multiply

The contents of the accumulator are multiplied by the contents of the addressed memory location, and the product is left in the accumulator. The word in memory remains intact.

The address may be modified by indexing.

An address of 3000 will cause the instruction to square the contents of the accumulator, leaving the result in the accumulator.

Op. Code	Address	Index Tag	Meaning
$\boxed{/}$	0000 through 3000	Yes	Divide

The contents of the accumulator are divided by the contents of the addressed memory location, and the quotient is left in the accumulator. The word in memory remains intact.

The address may be modified by indexing.

An address of 3000 will cause the contents of the accumulator to be divided by the contents of the accumulator, leaving a value of 1 in the accumulator. NOTE: if the divisor is zero, an overflow will be generated, causing RIP-3000 to print an error note on the typewriter and halt.

Op. Code	Address	Index Tag	Meaning
$\boxed{v}$	0000 through 3000	Yes	Inverse Divide

The contents of the addressed memory location are divided by the contents of the accumulator, and the quotient is left in the accumulator. The word in memory remains intact.

The address may be modified by indexing.

An address of 3000 will produce a quotient of 1 in the accumulator. NOTE: If the divisor is zero, an overflow will be generated, causing RIP-3000 to print an error note on the typewriter and halt.

Op. Code	Address	Index Tag	Meaning
$\boxed{s}$	0000 through 3000	Yes	Store accumulator

The contents of the accumulator are stored in the addressed memory location, replacing the original contents of the memory location. The word in the accumulator remains intact.

The address may be modified by indexing.

An address of 3000 will cause the instruction to do nothing, unless modified by an index register.

EXAMPLE

Assume the following:

Location 2000 contains the value, A.
Location 2001 contains the value, B.
Location 2002 contains the value, C.

The value, AB/C + A/(B+C) will be computed by the following sequence of instructions leaving the result in the accumulator.

Location	Operation	Address	Index	Remarks
400	]	2000		A to accumulator
401	.	2001		AB to accumulator
402	/	2002		AB/C in accumulator
403	s	2005		Store temporarily in 2005
404	]	2001		B to accumulator
405	+	2002		B+C in accumulator
406	v	2000		A/(B+C) in accumulator
407	+	2005		AB/C + A/(B+C) in accumulator

Op. Code	Address	Index Tag	Meaning
↑	0000 through 3000	Yes	Power

The contents of the accumulator are computed to the power specified by the contents of the addressed memory location, and the result is left in the accumulator. The word in memory remains intact. The original contents of the accumulator must be greater than zero.

The address may be modified by indexing.

An address of 3000 will cause the contents of the accumulator to be computed to the power indicated by the original contents of the accumulator.

Op. Code	Address	Index Tag	Meaning
r	0000 through 3000	Yes	Square root

The square root of the contents of the addressed memory location is computed, and the result is left in the accumulator. The word in memory remains intact.

The address may be modified by indexing.

An address of 3000 will cause the square root of the accumulator contents to be extracted and left in the accumulator.

Op. Code	Address	Index Tag	Meaning
z	None	No	Clear Accumulator to Zero

The contents of the accumulator are set to zero.

An address or index tag with this instruction is meaningless and, if entered, will be ignored.

Op. Code	Address	Index Tag	Meaning
p	None	No	Set Accumulator Positive

The sign of the accumulator contents, if negative, is set to positive. If the sign is already positive, the instruction does nothing.

The magnitude of the value in the accumulator is not changed. An address or index tag with this instruction is meaningless and, if entered, will be ignored.

Op. Code	Address	Index Tag	Meaning
$\boxed{\text{n}}$	None	No	Change Accumulator Sign

The sign of the accumulator, if negative, is set to positive. The sign of the accumulator, if positive, is set to negative.

The magnitude of the value in the accumulator is not changed. An address or index tag with the instruction is meaningless and, if entered, will be ignored.

EXAMPLE

Assume the following:

Location 1500 contains the value, A.
Location 1501 contains the value, B.
Location 1502 contains the value, C.
Location 1503 contains the exponent, n.

The following sequence of instructions will compute the value, $-(A^n)/\sqrt{B} - |C|$, store the result in location 1504, and set location 1503 to zero.

Location	Operation	Address	Index	Remarks
2000	]	1502		C to Accumulator
2001	p			$\lvert C \rvert$ in Accumulator
2002	s	2100		Store temporarily in 2100
2003	r	1501		$\sqrt{B}$ to Accumulator
2004	s	2101		Store temporarily in 2101
2005	]	1500		A to Accumulator
2006	↑	1503		A^n in Accumulator
2007	n			$-(A^n)$ in Accumulator
2008	/	2101		$-(A^n)/\sqrt{B}$ in Accumulator
2009	-	2100		$-(A^n)/\sqrt{B} - \lvert C \rvert$ in Accumulator
2010	s	1504		Store in 1504
2011	z			Zero Accumulator
2012	s	1503		Zero location 1503

TRANSFER INSTRUCTIONS

Transfer instructions provide a control over the sequence in which instructions are executed. Instructions will be executed in the linear sequence in which they are stored in memory until a transfer instruction is encountered. At this point, the sequence is interrupted, and control is transferred to a selected location in memory where processing is to be resumed. A transfer instruction may be unconditional, or it may be dependent upon the state of the value in the accumulator or in a particular memory location.

CAUTION: The index tag must not be used with t, w, or m.

Op. Code	Address	Index	Meaning
t	0000 through 3000	No	Transfer Control

Control is unconditionally transferred to the instruction in the addressed memory location.

An address of 3000 will cause a transfer to the accumulator. In this case, if the accumulator contains anything except a "t" or "w" instruction, control of the program will be lost.

The index tag must not be used. If an index tag is entered, program control will be lost.

Op. Code	Address	Index	Meaning
w	0000 through 3000	No	Wait and Transfer

Computer operation will halt upon encountering this instruction. Moving the COMPUTE switch on the control panel to "halt" and back to "compute" will cause operation to be resumed, starting with the instruction contained in the addressed memory location.

An address of 3000 will cause a transfer to the accumulator. In this case, if the accumulator contains anything except a "t" or "w" instruction, control of the program will be lost.

The index tag must not be used. If an index tag is entered, program control will be lost.

Op. Code	Address	Index	Meaning
b	0000 through 3000	Yes	Conditional Branch

Control will be transferred subject to the condition of the word in the addressed memory location or the accumulator location, 3000. If the word is less than zero, no branch will occur and the next instruction in sequence will be executed. If the word is equal to zero, the next instruction in sequence will be skipped. If the word is greater than zero, the next two instructions in sequence will be skipped.

The address of the word on which the branch is conditional may be modified by indexing.

EXAMPLE

Assume the following: Location 2500 contains the value, A.
Location 2510 contains the value, B.

The instruction sequence will follows will compute A - B. If the result is less than zero, it will store it in location 1000. If the result equals zero, it will store it in location 1001. If the result is greater than zero, it will store it in location 1002. When finished, it will halt. Manual restart will transfer control to location 1500.

Location	Operation	Address	Index	Remarks
0	j	2500		A to Accumulator
1	-	2510		A - B in Accumulator
2	b	3000		Skip to 4 if zero, to 5 if > zero.
3	t	7		Value< zero, transfer to 7.
4	t	9		Value=zero, transfer to 9.
5	s	1002		Store value> zero.
6	w	1500		Wait and transfer to 1500.
7	s	1000		Store value < zero.
8	w	1500		Wait and transfer to 1500.
9	s	1001		Store value = zero.
10	w	1500		Wait and transfer to 1500.

Op. Code	Address	Index Tag	Meaning
m	0000 through 3000	No	Mark and Transfer

The location of this instruction is marked, and control is unconditionally transferred to the instruction contained in the addressed memory location. This instruction is used to call a closed subroutine which will, when finished, return control to the instruction following the "m" instruction. An address of 3000 will cause a transfer to the accumulator. In this case, if the accumulator contains anything except a "t" or "w" instruction, control of the program will be lost.

The index tag must not be used. If an index tag is entered, program control will be lost.

Op. Code	Address	Index Tag	Meaning
$\boxed{e}$	None	No	Subroutine Exit

Control is unconditionally transferred to the instruction contained in the memory location following the last executed "m" instruction. This instruction is used to exit from a closed subroutine which was entered from an "m" instruction.

An address or index tag, if entered, will be ignored.

EXAMPLE

Assume the following: Locations 2000 through 2002 contain three values of A.
Location 2005 contains the exponent, n.

The following instruction sequence will compute A^n-2A for the three values of A, and store the results in 2100 through 2102.

Location	Operation	Address	Index	Remarks
1000	]	2000		A_1 to Accumulator
1001	m	1010		Mark and Transfer to subroutine.
1002	s	2100		Store $(A^n-2A)_1$
1003	]	2001		A_2 to Accumulator
1004	m	1010		Mark and transfer to subroutine
1005	s	2101		Store $(A^n-2A)_2$
1006	]	2002		A_3 to Accumulator
1007	m	1010		Mark and transfer to subroutine
1008	s	2102		Store $(A^n-2A)_3$
1009	w	1000		Wait and transfer to beginning of program

Location	Operation	Address	Index	Remarks
1010	s	2010		SUBROUTINE -- Store A temporarily.
1011	+	3000		2A in Accumulator .
1012	s	2011		Store 2A temporarily.
1013	]	2010		A to Accumulator.
1014	↑	2005		A^n in Accumulator .
1015	-	2011		A^n - 2A in Accumulator.
1016	e			Exit to location 1002, 1005, or 1008 .

USING THE INDEX REGISTERS

It is often desirable in programming to execute the same sequence of instructions repetitively for a given number of times before proceeding to the next portion of the program. For example, it may be necessary to perform the same operations on each entry in a sequentially stored data table. Index registers provide an effective means for controlling these program loops. They serve the dual purpose of address modification and counting.

Nine index registers are maintained by RIP-3000, thereby providing simultaneous control of nine processing functions.

Any instruction which makes reference to data stored in memory may be modified by indexing. Indexing is specified for an instruction by entering an index register number (1 through 9) following the space(s) terminating an instruction address. The contents of the specified index register are subtracted from the original address before execution of the instruction to produce the effective address. Therefore, in referencing a table of values, the address portion of the instruction is normally one greater than the highest address of the table.

Op. Code	Address	Index Tag	Meaning
1	0000 through 3000	Yes	Load Index

The index register indicated by the tag will be loaded with the address portion of the instruction if it is in the range 0000 through 2999. The index setting comes from the instruction word itself -- not from memory.

An address of 3000 will cause the absolute value of the integral portion of the accumulator contents to be loaded into the index register. If this number exceeds 4095, it will not be loaded correctly.

Op. Code	Address	Index Tag	Meaning
$\boxed{x}$	0000 through 3000	Yes	Decrement Index by One and Transfer

The contents of the index register specified by the tag will be decremented by one. If the index value is not now zero, or less than zero, control will be transferred to the instruction contained in the addressed memory location. Otherwise, the next instruction in sequence will be executed.

An address of 3000 should not be used unless the accumulator contains a "t" or "w" instruction.

Op. Code	Address	Index Tag	Meaning
$\boxed{a}$	0000 through 30000	Yes	Decrement Index by Address

The contents of the specified index register will be decremented by the address portion of the instruction if it is in the range 0000 through 2999. An address of 3000 will cause the index contents to be decremented by the absolute value of the integral portion of the accumulator contents.

This instruction will <u>not</u> cause a transfer of control.

EXAMPLE

The following instruction sequence will compute the sum of the contents of locations 500 through 599, and halt with the sum in the accumulator.

Location	Operation	Address	Index	Remarks
0	1	100	8	Set index 8 = 100 .
1	z			Clear accumulator to zero .
2	+	600	8	Add one number.
3	x	2	8	Decrement index 8. If > 0, return to 2.
4	w	1000		Halt. Resume at location 1000.

EXAMPLE

Assume the following:

Locations 1000 through 1019 contain a table in which every fifth word beginning with 1000 represents a value of A.

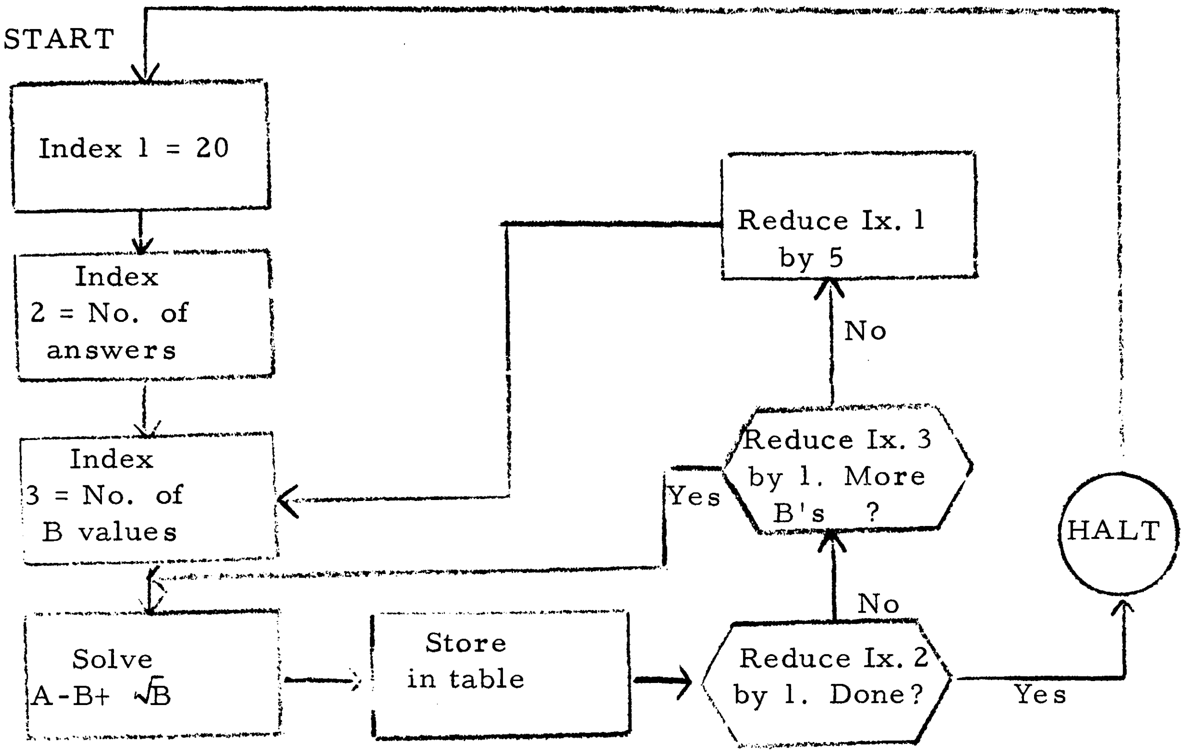
Location 1269 is the final location of a variable length table of values of B. Maximum number of locations = 250.

Location 999 contains the number of B values in the above table.

The sequence of instructions below will compute A - B + $\sqrt{B}$ for the four values of A and all values of B. The results will be stored in a table ending at location 2269.

Location	Operation	Address	Index	Remarks
500	l	20	1	Set index 1 = 20
501	]	999		Number of B's to Accumulator
502	+	3000		Double number of B's
503	+	3000		Four times the number of B's
504	l	3000	2	Set index 2 = number of answers
505	]	999		Number of B's to accumulator
506	l	3000	3	Set index 3 = number of B's
507	r	1270	3	$\sqrt{B_n}$ to accumulator
508	+	1020	1	$A_n + \sqrt{B_n}$ in accumulator
509	-	1270	3	$A_n - B_n + \sqrt{B_n}$ in accumulator
510	s	2270	2	Store in table ending at 2269
511	x	513	2	Decrement index 2 by 1. If > 0, go to 513
512	w	500		Halt. Restart at 500
513	x	507	3	Decrement index 3 by 1. If >0, return to 507
514	a	5	1	Decrement index 1 by 5
515	t	505		Return to 505

A flow diagram will be helpful in interpreting the above instruction sequence.



INPUT/OUTPUT INSTRUCTIONS

Op. Code	Address	Index Tag	Meaning
i	0000 through 3000	Yes	Input

A data value will be input and stored in the addressed memory location and in the accumulator. An address of 3000 will cause the value to be stored in the accumulator only.

The address may be modified by indexing.

The absolute value of the input must be less than 1.7×10^{38}.

The absolute value of the exponent must be less than 39.

No more than 11 significant digits may be entered.

The number is entered in the following form:

1. Sign of the number (may be omitted if positive).

2. A decimal number (a decimal point may be included anywhere in the number).

3. Sign of the exponent, if any.

4. An exponent (optional). The power of 10 by which the number is to be multiplied.

5. A terminate code (tab, carriage return, comma, space or stop code).

Op. Code	Address	Index Tag	Meaning
$\boxed{\text{o}}$	0000 through 3000	Yes	Output

The data value contained in the addressed memory location will be printed out. An address of 3000 will cause the value contained in the accumulator to be output. In both cases, the original accumulator contents will be preserved.

The address may be modified by indexing.

The absolute value of the number to be output may not exceed 8.5 x 10^{37}.

The number will be output in the following form:

1. Sign of number.

2. The number, consisting of one integral digit, a decimal point and eight fractional digits.

3. Space.

4. Sign of exponent.

5. A two-digit exponent indicating the power of 10 by which the number is to be multiplied.

EXAMPLE

The following sequence of instructions will input two numbers, multiply them together, and output their product.

Location	Operation	Address	Index	Remarks
2000	i	2100		Input number and store in 2100.
2001	i	3000		Input number to accumulator
2002	.	2100		Multiply by number in 2100.
2003	o	3000		Output product
2004	w	2000		Halt. Return to 2000

Assume the inputs to be: -3. 25 C. R.
. 25+3 C. R.
The output will be: -8. 12500000 +02

Op. Code	Address	Index Tag	Meaning
f	1 through 3	Yes	Format Control

This instruction provides the facility for controlling typed format by generating the typewriter command codes for space, tab, and carriage return.

An address of 1 means "space".
An address of 2 means "tab".
An address of 3 means "carriage return".

The index registers have no significance with this instruction.

The index tag field is used to specify how many spaces, tabs, or carriage returns are to be output, and must be in the range 1 through 9. No tag is equivalent to a tag of one.

Op. Code	Address	Index Tag	Meaning
h	1 through 26	No	Print Heading

Alphanumeric information may be output using this instruction. The address specifies the number of words in sequence, following this instruction, which contain the information to be printed. The alphanumeric words contain five characters each, but characters are entered contiguously without the usual terminating carriage return. The number of characters must be a multiple of five. Unused character positions must be filled with blanks or spaces. Therefore, if a heading consisting of 17 characters is to be printed, the instruction address must be 4 and three blanks must be entered following the last character. Typewriter command codes (upper case, lower case, etc.) are recognized as legitmate characters. Good practice requires that the typewriter be left in lower case following an alphanumeric output.

The maximum length output for one instruction is 130 characters (26 words). When the last character of a heading has been input, RIP-3000 will automatically return the carriage.

Op. Code	Address	Index Tag	Meaning
$\boxed{j}$	0000 through 3000	Yes	Step Counter and Print

The number contained in the addressed memory location is incremented by one, and the absolute value of the result is output as a four-digit integer followed by a space. An address of 3000 will cause the accumulator to be incremented and printed.

The address may be modified by indexing.

EXAMPLE

Assume the following: Locations 1000 through 1004 contain five values of A.

The following instruction sequence will input five values of B, add the corresponding A and B values, and output the results with a heading and in the format shown following the coding:

Location	Operation	Address	Index	Remarks
2000	l	5	1	Set index 1 = 5 .
2001	i	1010	1	Input B's to 1005 through 1009.
2002	x	2001	1	Decrement index 1. If > 0, return to 2001.
2003	h	3		Print next 3 words as heading .
2004	tab u/c A tab B			
2005	tab A l/c + u/c			Alphanumeric heading .
2006	B l/c c. r. blank blank			
2007	z			Set accumulator = zero.
2008	s	1100		Store zero in 1100.
2009	l	5	1	Set index 1 = 5.
2010	j	1100		Step counter and print sequence number.
2011	f	2		Output tab .
2012	]	1005	1	A to accumulator .
2013	+	1010	1	A + B in accumulator .
2014	o	1005	1	Print A value.
2015	f	2		Output tab .
2016	o	1010	1	Print B value .
2017	f	2		Output tab .
2018	o	3000		Print A + B value .
2019	f	3		Output carriage return .
2020	x	2010	1	Decrement index 1. If > 0, return to 2010 .
2021	f	3	3	Output 3 carriage returns.
2022	w	2000		Halt. Return to 2000.

Assuming the typewriter tab stops properly set, the output of the above sequence will be in the following format:

```
        A                       B                       A+B
0001    ± d. ddddddd ± ee       ± d. ddddddd ± ee       ± d. ddddddd ± ee
0002    ± d. ddddddd ± ee       ± d. ddddddd ± ee       ± d. ddddddd ± ee
0003    ± d. ddddddd ± ee       ± d. ddddddd ± ee       ± d. ddddddd ± ee
0004    ± d. ddddddd ± ee       ± d. ddddddd ± ee       ± d. ddddddd ± ee
0005    ± d. ddddddd ± ee       ± d. ddddddd ± ee       ± d. ddddddd ± ee
```

where d represents a digit of the numerical value and ee represents the exponent.

MISCELLANEOUS INSTRUCTIONS

Op. Code	Address	Index Tag	Meaning
Space	None	No	No Operation

The instruction will occupy a memory location, but will have no effect on program operation.

It is often desirable in checking out a program to insert certain instructions which will be unnecessary in the final version. The "space" instruction may be used to replace these unneeded instructions.

Op. Code	Address	Index Tag	Meaning
u	0000 through 3000	Yes	Exchange

The contents of the addressed memory location are exchanged with the contents of the accumulator.

An address of 3000 will cause the instruction to do nothing, unless modified by an index register.

Op. Code	Address	Index Tag	Meaning
g	1 through 12	No	Execute Subroutine

This instruction calls upon one of a set of mathematical subroutines. Addresses 1 through 7 specify subroutines which are incorporated in the RIP-3000 program. Addresses 8 through 12 are available to call additional subroutines, tailored to the user's individual requirements. (See Appendix 1.)

The argument must be in the accumulator when the "g" instruction is executed. The result will be left in the accumulator.

The subroutines available in RIP-3000 are as follows:

1. Sine
2. Cosine
3. Log base 10
4. Log base e
5. 10^X
6. e^X
7. Arctangent

Op. Code	Address	Index Tag	Meaning
k	0000 through 2999	No	Store a Constant

The numerical constant immediately following this instruction will be stored in the addressed memory location. The "k" instruction itself will not occupy a memory location. It serves only to direct RIP-3000 to store the constant during the input processing phase.

The constant is entered in the following form:

1. Sign of the number may be omitted if positive.
2. A decimal number (a decimal point may be included anywhere in the number).
3. Sign of the exponent, if any.
4. An exponent (optional).
5. A terminate code (tab, carriage return, comma, space, or stop code).

NOTE: An address of 3000 causes the instruction to do nothing.

An index tag, if entered, will be ignored.

Op. Code	Address	Index Tag	Meaning
d	None	No	Debug

This instruction causes the program to be executed in the trace mode, beginning with the instruction immediately following the "d" instruction. Following the execution of each instruction in the trace mode, the location counter and the instruction will be printed out. If the instruction contains an index tag, the absolute value of the specified index register will be printed as a four-digit integer. If the instruction modifies the accumulator, the accumulator contents will be printed in the data output format specified for the "o" instruction.

An address or index tag with this instruction is meaningless, and if entered, will be ignored.

NOTE: Input and output instructions (i, o, f, h, and j) are not traced.

Op. Code	Address	Index Tag	Meaning
$\boxed{q}$	None	No	Quit Debugging

This instruction will terminate trace mode operation.

An address or index tag with this instruction is meaningless, and, if entered, will be ignored.

EXAMPLE

The following instruction sequence will input 100 values of A. It will compute -3.75 cos A + A^2 for each entry and store it in a table beginning at location 1000. The portion of the program between the "d" and "q" instructions will be executed in the trace mode. When 100 values are processed, the "d" instruction is replaced by a "space" instruction, so that subsequent executions will not be traced.

Location	Operation	Address	Index	Remarks
2000	l	100	1	Set index 1 = 100.
2001	i	1500		A to accumulator and 1500
2002	g	2		Go to cosine subroutine.
	k	1501		Put constant -3.75 in 1501.
	-3.75			
2003	d			Start tracing.
2004	.	1501		-3.75 cos A in accumulator..
2005	u	1500		A to accumulator; -3.75 cos A to 1500
2006	.	3000		A^2 in accumulator.
2007	+	1500		-3.75 cos A + A^2 in accumulator.
2008	s	1100	1	Store in table beginning at 1000
2009	q			Stop tracing
2010	x	2001	1	Decrement index. If >0, return to 2001.
2011	]	2014		"Space" instruction to accumulator.
2012	s	2003		Replace "d" with "space".
2013	w	2500		Halt. Resume at 2500
2014	space			No operation,

IV. OPERATING PROCEDURES

THE BASIC RECOMP III COMPUTER

The basic RECOMP III computer consists of a 4096-word memory unit, an operator's control console, and a Flexowriter for input and output.

The programmer using RIP-3000 need be concerned only with the control console and the Flexowriter. It is sufficient that he understand the procedure for entering information into the computer and for communicating with the RIP-3000 program.

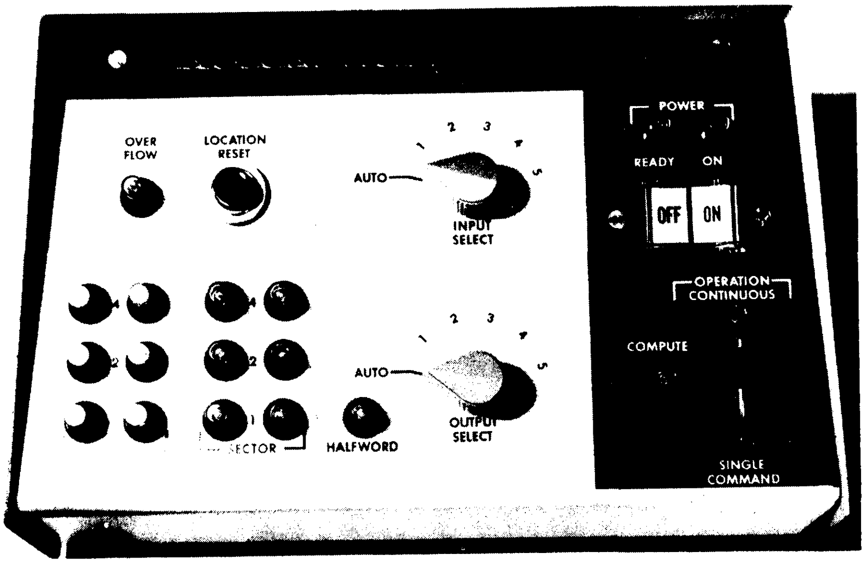


Figure 1. The RECOMP III Control Console

The RECOMP III Control Console contains the following switches and indicators:

POWER ON/OFF SWITCH — A two position switch to turn computer power on or off.

POWER ON INDICATOR — A light which indicates computer power is on.

READY INDICATOR — A light which comes on approximately 45 seconds after computer power is turned on. When this light is on, the memory disk has reached proper speed and the computer is ready for operation.

COMPUTE SWITCH

A three-position switch to start or stop program execution. In the CONTINUOUS position, the computer will operate continuously under program control. The HALT position will stop computer execution of a program. The SINGLE COMMAND position is of no concern to the RIP-3000 programmer.

COMPUTE INDICATOR

A light which is on when the computer is executing program instructions.

INPUT SELECT SWITCH

A six-position rotary switch to select an input device. In the AUTO position, the input device is selected by the program. RIP-3000 is programmed to input via the typewriter keyboard. Positions 1 through 5 select an alternate input device, over-riding program selection.

OUTPUT SELECT SWITCH

A six-position rotary switch to select an output device. In the AUTO position, the output device is selected by the program. RIP-3000 is programmed to output via the typewriter keyboard. Positions 1 through 5 select an alternate output device, over-riding program selection.

LOCATION RESET SWITCH

A momentary switch to set the computer's location counter to zero. For the RIP-3000 programmer, this switch is of significance only in transferring to the Load/Start Routine.

OVERFLOW INDICATOR

This indicator is of no concern to the RIP-3000 programmer.

LOCATION INDICATOR LIGHTS This display is of no concern to the RIP-3000 programmer.

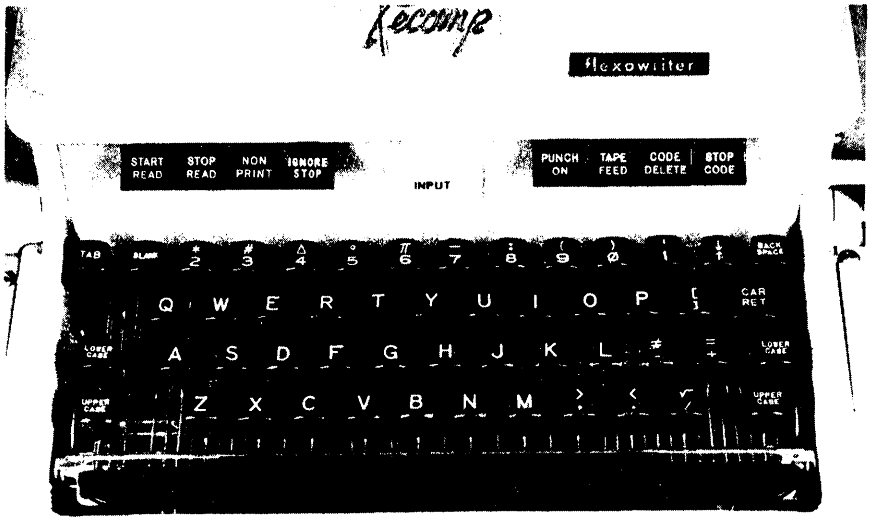


Figure 2. The Flexowriter Keyboard

THE FLEXOWRITER

The Flexowriter provides the basic input/output facilities for the RECOMP III computer. Information may be input via the typewriter keyboard or the paper tape reader. Information may be output via the typewriter or the paper tape punch.

Those switches and indicators of concern to the RIP-3000 programmer are as follows:

ON/OFF SWITCH A two-position switch to turn Flexowriter power on or off.

LOCAL/COMPUTE SWITCH A two-position switch to select off-line or on-line operation. This switch must be in the COMPUTE position to process or execute a RIP-3000 program.

INPUT INDICATOR An indicator which is illuminated when the computer is calling for an input from the Flexowriter keyboard.

The bank of eight switches above the keyboard are of no concern to the RIP-3000 programmer and should be in the raised or "off" position.

For paper tape input and output the tape should be threaded through the read and punch stations as indicated in Figure 3.

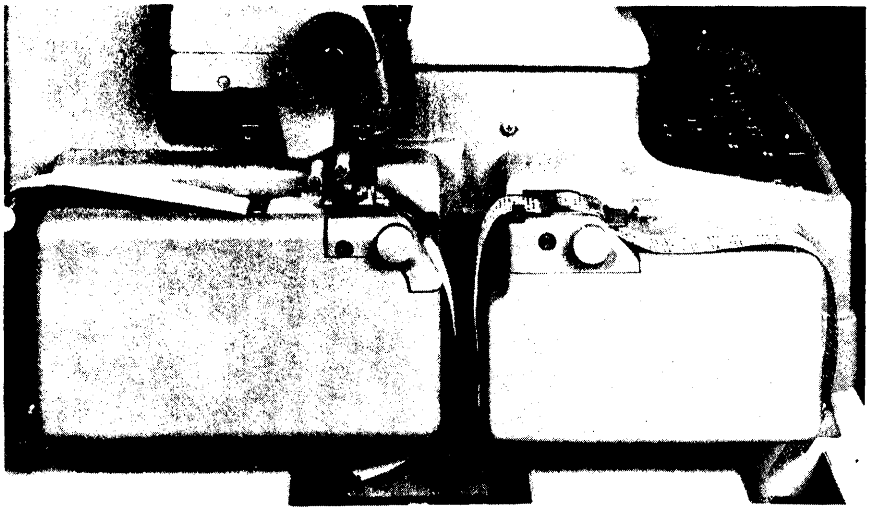


Figure 3. Flexowriter Tape Punch and Tape Reader

RIP-3000 OPERATING PROCEDURES

The Load/Start Routine must be in the computer in locations 0000_8 through 0177_8 to load RIP-3000 or to process a program written in the RIP-3000 language. This may be verified by depressing the LOCATION RESET switch, moving the COMPUTE switch to CONTINUOUS, and depressing the "q" key on the typewriter. If Load/Start is not intact, a slash (/) will be printed. Appendix 2 defines the procedure for loading the Load/Start Routine (R3P-1).

LOADING THE RIP-3000 TAPE

The procedure for loading the RIP-3000 tape is as follows:

1. Place the tape in the reader.
2. Position the INPUT SELECT switch to read from tape (position 2).
3. Depress the LOCATION RESET switch.
4. Position the COMPUTE switch to CONTINUOUS. The tape will load automatically and the computer will halt. If an error is detected in loading the tape, a slash (/) will be printed.

NOTE: A quick check of RIP-3000 in memory may be made by loading the quick check routine on the front of the RIP-3000 tape. When reading stops, turn the INPUT SELECT switch to AUTO, then move the COMPUTE switch to HALT and back to CONTINUOUS. If RIP-3000 is not intact in memory, three slashes (///) will be printed and control will be returned to the Load/Start Routine. If RIP-3000 is in memory, the routine will transfer to it.

PROCESSING A RIP LANGUAGE PROGRAM (TYPEWRITER INPUT/OUTPUT)

The procedure for processing a RIP language program (typewriter input/output) is as follows:

1. Position the INPUT SELECT switch to AUTO.
2. Move the COMPUTE SWITCH to HALT, press the LOCATION RESET button, and move the COMPUTE switch back to CONTINUOUS. The INPUT light will come on.
3. Type s15000 followed by a carriage return to transfer to RIP-3000. When the Flexowriter INPUT light comes on, RIP-3000 is ready to load the first instruction in location 0000.

4. Type the location of the first instruction, if not 0000, terminated by a space, comma, or carriage return.

5. Type the operation code followed by one or more spaces, if desired, or by a carriage return or comma if no address is required.

6. Type the address, in decimal, followed by one or more spaces. If no index tag is required, follow the address with a carriage return or comma.

7. Type the index tag followed by a carriage return or comma.

8. Repeat Steps 5 through 7 for each instruction to be stored in sequence. A new starting location may be entered for an instruction sequence at any time RIP is calling for an instruction.

INPUT ERRORS

The procedure for handling input errors is as follows:

1. A non-existent operation code, an address over 3000, or an index tag over 9 will cause "bad" to be printed on the typewriter. The instruction must now be re-entered from the beginning.

2. A legal address which is determined to be in error before the terminating space, carriage return, or comma may be corrected by typing a slash (/) followed by the correct address.

3. An instruction which has been accepted and processed by RIP-3000 may be changed by typing the location, followed by the required instruction.

4. If the RIP-3000 location counter is advanced beyond 2999, it will return to 0000 and "0000" will be printed on the typewriter. The program will not operate properly in this event, unless location 2999 contains a transfer instruction.

DUMPING THE PROCESSED PROGRAM

The procedure for dumping the processed program is as follows:

1. Position the COMPUTE switch on the console to HALT.

2. Depress the LOCATION RESET switch.

3. Position the COMPUTE switch to CONTINUOUS.

4. Transfer to the dump routine by typing s20000 followed by a carriage return if program is to be dumped on tape.

5. Transfer to the dump routine by typing s20500 by a carriage return if program is to be dumped symbolically on the typewriter.

6. Enter the first location to be dumped, followed by a space, comma, or carriage return.

7. Enter the last location to be dumped, followed by a space, comma, or carriage return.

8. Additional blocks of memory may be dumped on tape by repeating Steps 4, 6, and 7.

9. Additional blocks of memory may be typed symbolically by repeating Steps 6 and 7.

RIP PROGRAM OPERATION

The procedure for operating a RIP program is as follows:

1. Load the RIP-3000 tape if not already in memory.

2. Load the program tape in the same manner if not already in memory.

3. Position the COMPUTE switch to HALT.

4. Turn the INPUT SELECT switch to AUTO.

5. Depress the LOCATION RESET switch.

6. Position the COMPUTE switch to CONTINUOUS.

7. When the INPUT light comes on, type s15000 followed by a carriage return to transfer to RIP-3000.

8. Type a "c" followed by the starting location of the program and a space, comma, or carriage return. The program will now operate automatically.

9. If any arithmetic operation produces a result which overflows the capacity of the accumulator, or if a division by zero is encountered, an error note will be printed giving the location of the instruction which generated the error.

APPENDIX 1

INCORPORATION OF SUBROUTINES

The "g" instruction in the RIP-3000 language is used to call one of a set of mathematical function subroutines. Seven such subroutines are built into RIP-3000 and are specified by "g" addresses 1 through 7. Provision has been made for the user to add from one to five subroutines specified by "g" addresses 8 through 12.

This appendix assumes a knowledge of RECOMP III machine language. Addresses are octal and refer to the computer memory.

The procedure for adding subroutines to RIP-3000 is as follows:

1. Code the subroutine to be added in such a way as to save the L loop and index register on entry and restore them prior to exit.

2. The subroutine should assume the RIP accumulator to be in the R register on entry. The desired RIP accumulator contents should be in the A register on exit.

3. The subroutine must exit to computer location 7763. 1.

4. The coding for the subroutine must be located between 2070_8 (RIP location 0000) and location 7757_8 (RIP location 2999^8). Note the RIP memory locations occupied by the subroutine. A RIP language program must not be stored in, or refer to, these addresses.

5. In computer location 1627_8, 1630_8, 1631_8, 1632_8, or 1633_8 (corresponding to "g" addresses 8 through 12) insert the following word in command format:

   -xxxxy00-0003300

   where xxxx specifies the starting address of the subroutine, and the left-most bit of y is the half-word bit. For example, if the subroutine starts in computer location 3145. 1, the word inserted would be -3145400-0003300.

6. If "g" addresses 10, 11, or 12 are to be used, insert the following word in command format into computer location 2057_8:

   +0017370-0000020

   This change is necessary to output a two-digit "g" address during tracing or symbolic memory dump.

APPENDIX 2

LOADING THE LOAD/START ROUTINE

The Load/Start Routine must be in memory locations 0000_8 through 0177_8 in order to load or use RIP-3000. The beginning of the Load/Start tape has a self-loading bootstrap routine which requires the following manual procedure:

1. Place the Load/Start tape in the reader.
2. Position the INPUT SELECT switch to AUTO.
3. Depress the LOCATION RESET switch and hold it down during steps 4 and 5.
4. Position the COMPUTE switch to CONTINUOUS.
5. When reading stops, position the COMPUTE switch to HALT.
6. Release the LOCATION RESET switch.
7. Position the COMPUTE switch to CONTINUOUS. The remainder of the tape will now be loaded by the bootstrap routine.

NOTE: A quick check can be made to determine whether Load/Start is intact in memory.

1. Position the COMPUTE switch to HALT.
2. Depress the LOCATION RESET switch.
3. Position the COMPUTE switch to CONTINUOUS.
4. Type "q" on the typewriter. If Load/Start is <u>not</u> intact, a slash (/) will be printed.

APPENDIX 3

LIST OF RIP-3000 OPERATIONS

Operation Code	Address	Index Tag	Meaning
]	0000 through 3000	Yes	Clear and Add
+	0000 through 3000	Yes	Add to Accumulator
-	0000 through 3000	Yes	Subtract from Accumulator
.	0000 through 3000	Yes	Multiply
/	0000 through 3000	Yes	Divide
v	0000 through 3000	Yes	Inverse Divide
s	0000 through 3000	Yes	Store Accumulator
↑	0000 through 3000	Yes	Power
r	0000 through 3000	Yes	Square Root
z	None	No	Clear Accumulator to Zero
p	None	No	Set Accumulator Positive
n	None	No	Change Accumulator Sign
t	0000 through 3000	No	Transfer Control
w	0000 through 3000	No	Wait and Transfer
b	0000 through 3000	Yes	Conditional Branch
m	0000 through 3000	No	Mark and Transfer
e	None	No	Subroutine Exit
l	0000 through 3000	Yes	Load Index
x	0000 through 3000	Yes	Decrement Index by One and Transfer
a	0000 through 3000	Yes	Decrement Index by Address
i	0000 through 3000	Yes	Input
o	0000 through 3000	Yes	Output
f	1 through 3	Yes	Format Control
h	1 through 26	No	Print Heading
j	0000 through 3000	Yes	Step Counter and Print

Operation Code	Address	Index Tag	Meaning
space	None	No	No Operation
u	0000 through 3000	Yes	Exchange
g	1 through 12	No	Execute Subroutine
k	0000 through 2999	No	Store a Constant
d	None	No	Debug
q	None	No	Quit debugging

Recomp III

PROGRAM NO. R3P-39

ORIG. DATE 18 November 1962

REV. DATE

PROGRAMMER G. Howell

PAGE 1 **OF** 5

PROGRAM TITLE: RIP-3000 (FLOATING POINT, MODIFIED)

1. INTRODUCTION

This program consists of a modified version of RIP-3000 (R3P-16) which can be used only on machines with floating point hardware. Please refer to the RIP Manual and the R3P-16 write-up for anything not mentioned in this write-up.

2. RESTRICTIONS

2.1 The computer must have floating point hardware.

2.2 The LOAD/START routine, R3P-1, must be in the computer.

3. USE

3.1 This version has two starting locations, 1500.0 or 1501.0. The start at 1501.0 is the same as the old 1500.0, and should be used when loading a symbolic tape. The start at 1500.0 causes the location of each command to be typed out prior to entering that command. If a new location is entered, it will be repeated. If a command is entered into location 2999, 0000 will be printed twice instead of once.

3.2 There is a decimal memory dump analogous to the program dump at 2050.0. It starts at 1300.0.

3.3 There are provisions for 24 g functions, with g 13 through g 24 in locations (machine octal) 1634 through 1647. The tape includes g 8 for fixed point output, g9 for matrix inversion and solution of simultaneous linear equations, and g 10, 11, and 12 for plotting. (See paragraphs 3.6 through 3.8).

3.4 There are 2 new commands, y and , (comma). Comma causes a transfer back to the RIP program input at 1500.0. To change the comma command to a transfer to location 1501.0 (which will allow RIP coding to call for program tapes) or to any location, change word 0450_8 to +51xxxx.x+777761.0, where xxxx.x is the location to which transfer is desired.

The y command, which may itself be tagged, causes the integral part of the accumulator to be added to the command addressed, if the command addressed is tagged. If it is not, the integral part of A will be subtracted. Example:

Assume A has -5.3 in it. In locations 1000 and 1001 are the commands:

```
1000 + 200 3
1001 s 300
```

The commands y 1000, y 1001 will cause these commands to become:

```
1000 + 195 3
1001 s 305
```

A y command must not refer to an f, g, or h command.

3.5 The constant command, k, has been modified to allow consecutive constants to be entered with one command. The command k ADDRESS works as always. The command k ADDRESS TAG calls for TAG constants to be entered consecutively starting at ADDRESS. TAG may be greater than 9. Thus;

k 100 50 calls for 50 constants to be entered into locations 100, 101, ..., 149.

PROGRAM TITLE: RIP-3000 (FLOATING POINT, MODIFIED)

3.5 (Continued)

If s1500.0 was used to start the input, the letter "k" and the location will print before each constant, as follows:

```
0100 s 1000   (any command)
0101 k75 3
k0075 -6
k0076 7
k0077 3+2
0101  (Since the k command takes no space, the
        location is still at 101).
```

or

```
0101 k75  (no tag)
k0075 -6
0101
```

3.6 The Fixed Point output, g 8, assumes that the number to be output is in the RIP accumulator. The format is controlled by loading index register 8 with LR, where L is the number of digits to the left and R to the right of the decimal point. For example, to get 3 digits to the left and 4 to the right, give 1 34 8. (See the write-up of R3P-44 for more details).

3.7 There are 3 plotter g - functions, g10, 11, and 12.

g10 Set Scale Factors and Original Location.
g11 Pen Up and Move (Traverse).
g12 Pen Down and Move (Plot).

1. When g11 or g12 are given as commands, the computer assumes that the coordinates of the point to which to move (X, Y) are stored in RIP 2725 and 2726 respectively.

PROGRAM TITLE: RIP-3000 (FLOATING POINT, MODIFIED)

3.7 (Continued)

2. Before any g11 or g12 commands are given, a g10 must be executed. This tells the routine where the plotter is now and how much 1 inch equals in X and Y. The initial position, (X_o, Y_o), must be stored in RIP 2721 and 2722 respectively, and the X and Y Scale Factors, X_s and Y_s, in RIP 2723 and 2724 respectively. The Scale Factors equal the number per inch/100. E.g., if 1 inch in the Y direction equals 750, $Y_s = 750/100 = 7.5$. It is not necessary that $X_s = Y_s$.

3.8 The g9 command will invert matrices or solve simultaneous linear equations as follows:

```
]   Key number
g 9
```

The form of the key number is ± LOCMMNN. where LOC is the RIP location of the first matrix element, MM (2 digits) is the number of rows, and NN (2 digits) is the number of columns. Matrices and constant columns are stored by column. If the key number is + the matrix will be inverted. If not, only simultaneous equations will be solved. It is possible to do both (if NN> MM). There may be more than one constant column to a set of equations. In all cases, the determinant of the matrix will be in the RIP accumulator on return.

Examples of key numbers: (all data starts in 100):

1. Invert 5 x 5 matrix: +1000505.

2. Solve 3 simultaneous equations in 3 unknowns with 6 constant columns: -1000309.

PROGRAM TITLE: RIP-3000 (FLOATING POINT, MODIFIED)

3.8 (Continued)

3. Solve 10 simultaneous equations in 10 unknowns with one constant column and also invert: +1001011.

4. Only the determinant of a 4 x 4 matrix is desired: -1000404. (Solution of a 4 x 4 system of equations with no constant columns; faster than inversion).

All data is stored by column followed by constant columns, if any. All answers replace data; i.e., the original matrix and constant columns are destroyed.

Timing: Inversion: $\sim \frac{(MM)^3}{15}$ seconds

Simultaneous Equations: $\sim \frac{(MM)(NN)}{1.5}$ seconds

For further details please refer to the write-up of R3S-039.

FUNCTION	NAME	SPACE OCCUPIED	
		OCTAL	RIP
g9	Matrix Inversion and Simultaneous Equations	7000-7327	2504-2719
g10, 11, 12	Plotter	7331-7513	2721-2835
g8	Fixed Point Output	7514-7717	2836-2967
	Quick Check	7720-7757	2968-2999

RECOMP III TECHNICAL BULLETIN NO. 5

TITLE: SEQUENTIAL OUTPUTTING OF DATA USING RIP.

PURPOSE: To show how a program may be written within RIP to output data sequentially.

EFFECTIVE DATE: 1 March 1962

CONTENTS:

AUTHOR: L. Laubscher

1. INTRODUCTION

1.1 The Users of RIP often find it convenient or necessary to output partial results or the contents of certain addresses in the RIP memory after the run, or during the debugging of a program. Three programs written in RIP to accomplish this are described below.

(1) The first program described on page 2 will input a 4 digit RIP address, type out the contents of this address and return to input another 4 digit address.

(2) The program described on page 3 will input a RIP address, and will output the contents of each location beginning with that address and continuing to address 2999.

(3) The program described on page 4 will input two RIP addresses and then will output the contents of each address beginning with the first and continuing up to and including the second.

RECOMP III INTERPRETIVE PROGRAM (RIP-3000)

TITLE EXAMPLE A PROGRAMMER ______ DATE ______ Page ___ of ___

ARITHMETIC			FUNCTIONS			INDEX		FORMAT-HEADING	
]	Clear & Add	(W) → A	g	W Indicates:		l	Load Index with W or (A)	f	Format; W Indicates:
+	Add	(A) + (W)		1-Sine	Sin (A)	a	Decr. Index by W or (A)		1 - Space
—	Subtract	(A) — (W)		2-Cosine	Cos (A)	x	Transfer on Index		2 - Tab
·	Multiply	(A) · (W) or (A) · (A)		3-Log (10)	Log_{10} (A)		TRANSFERS		3 - Carriage Return
/	Divide	(A) ÷ (W)		4-Log (e)	Ln (A)	t	Transfer to W	h	Heading Follows
s	Store	(A) → W		5-Expon. (10)	$10^{(A)}$	b	Branch; (A) or (W) < 0, = 0, > 0		MISCELLANEOUS
↑	Power	$(A)^{(W)}$		6-Expon. (e)	$e^{(A)}$	w	Wait (Halt & Transfer)	z	Clear (A) + 0 → A
r	Square Root	$\sqrt{(A)}$ or $\sqrt{(W)}$		7-Arctan	Tan^{-1} (A)		SUB-ROUTINES	p	Make (A) plus; + (A) → A
v	Inverse Divide	(W) ÷ (A)		INPUT-OUTPUT		m	Mark & Transfer to W	n	Change Sign (A); - (A) → A
u	Exchange	(A) ⇄ (W)	i	Input to A, or W & A		e	Exit Return to Last m		TRACING
Sp	No Operation		o	Output from A or W			CONSTANTS	d	Debug (Begin Tracing)
			j	Output 4 Digit Integer		k	Constants; Number Follows	q	Quit Tracing

LOCATION	OP	ADDRESS	I	REMARKS	STORAGE	
					LOCATION	DATA
2000	f	3	1	Return the carriage	2006	3000
2001	i	3000		Input the location		
2002	-	2006		Subtract 3000 from the location		
2003	l	3000	9	Load index 9 with the absolute value of the difference		
2004	o	3000	9	Output the contents of the location		
2005	t	2000		Transfer to 2000 to input another location		
	k	2006				

RECOMP III INTERPRETIVE PROGRAM (RIP-3000)

TITLE EXAMPLE B PROGRAMMER________ DATE________ Page____of____

ARITHMETIC		FUNCTIONS		INDEX		FORMAT-HEADING	
]	Clear & Add (W) → A	g	W Indicates:	l	Load Index with W or (A)	f	Format; W Indicates:
+	Add (A) + (W)		1-Sine Sin (A)	a	Decr. Index by W or (A)		1 - Space
—	Subtract (A) — (W)		2-Cosine Cos (A)	x	Transfer on Index		2 - Tab
·	Multiply (A) · (W) or (A) · (A)		3-Log (10) Log_{10} (A)	TRANSFERS			3 - Carriage Return
/	Divide (A) ÷ (W)		4-Log (e) Ln (A)	t	Transfer to W	h	Heading Follows
s	Store (A) → W		5-Expon. (10) $10^{(A)}$	b	Branch; (A) or (W) $<0, =0, >0$	MISCELLANEOUS	
↑	Power $(A)^{(W)}$		6-Expon. (e) $e^{(A)}$	w	Wait (Halt & Transfer)	z	Clear (A) + 0 → A
r	Square Root $\sqrt{(A)}$ or $\sqrt{(W)}$		7-Arctan Tan^{-1} (A)	SUB-ROUTINES		p	Make (A) plus; +(A) → A
v	Inverse Divide (W) ÷ (A)	INPUT-OUTPUT		m	Mark & Transfer to W	n	Change Sign (A); - (A) → A
u	Exchange (A) ⇄ (W)	i	Input to A, or W & A	e	Exit Return to Last m	TRACING	
Sp	No Operation	o	Output from A or W	CONSTANTS		d	Debug (Begin Tracing)
		j	Output 4 Digit Integer	k	Constants; Number Follows	q	Quit Tracing

LOCATION	OP	ADDRESS	I	REMARKS	STORAGE	
					LOCATION	DATA
2010	f	3	1	Return the carriage	2021	1
2011	i	3000		Input the location	2022	2999
2012	-	2021		Subtract 1	2023	Address-1
2013	s	2023		Store as the Address-1		
2014	-	2022		Subtract 2999		
2015	l	3000	9	Load difference into index 9		
2016	f	3	1	Return the carriage		
2017	j	2023		Increment the address by 1 and output		
2018	o	3000	9	Output the contents of the Address		
2019	x	2016	9	Reduce Index 9 by 1		
2020	w	2010		Wait to 2010		
	k	2021				
	k	2022				
	k	2023				

RECOMP III INTERPRETIVE PROGRAM (RIP-3000)

TITLE EXAMPLE C PROGRAMMER __________ DATE ________ Page ___ of ___

	ARITHMETIC		FUNCTIONS		INDEX		FORMAT-HEADING
]	Clear & Add (W) → A	g	W Indicates:	l	Load Index with W or (A)	f	Format; W Indicates:
+	Add (A) + (W)		1-Sine Sin (A)	a	Decr. Index by W or (A)		1 - Space
—	Subtract (A) — (W)		2-Cosine Cos (A)	x	Transfer on Index		2 - Tab
·	Multiply (A) · (W) or (A) · (A)		3-Log (10) Log_{10} (A)		TRANSFERS		3 - Carriage Return
/	Divide (A) ÷ (W)		4-Log (e) Ln (A)	t	Transfer to W	h	Heading Follows
s	Store (A) → W		5-Expon. (10) $10^{(A)}$	b	Branch; (A) or (W) < 0, = 0, > 0		MISCELLANEOUS
↑	Power $(A)^{(W)}$		6-Expon. (e) $e^{(A)}$	w	Wait (Halt & Transfer)	z	Clear (A) + 0 → A
r	Square Root $\sqrt{(A)}$ or $\sqrt{(W)}$		7-Arctan Tan^{-1} (A)		SUB-ROUTINES	p	Make (A) plus; + (A) → A
v	Inverse Divide (W) ÷ (A)		INPUT-OUTPUT	m	Mark & Transfer to W	n	Change Sign (A); - (A) → A
u	Exchange (A) ⇄ (W)	i	Input to A, or W & A	e	Exit Return to Last m		TRACING
Sp	No Operation	o	Output from A or W		CONSTANTS	d	Debug (Begin Tracing)
		j	Output 4 Digit Integer	k	Constants; Number Follows	q	Quit Tracing

LOCATION	OP	ADDRESS	I	REMARKS	STORAGE	
					LOCATION	DATA
2025	f	3	1	Return the carriage	2040	1
2026	i	3000		Input the first address	2041	2999
2027	-	2040		Subtract 1	2042	Address-1
2028	s	2042		Store as the address-1		
2029	-	2041		Subtract 2999		
2030	l	3000	9	Load the difference into Index 9		
2031	i	3000		Input the second address		
2032	-	2042		Obtain the number of address to be output		
				by subtracting the (first address-1)		
2033	l	3000	8	Load the number of address to be output		
				into Index 8		
2034	f	3	1	Return the carriage		
2035	j	2042		Output the address		
2036	o	3000	9	Output the contents of the address		
2037	x	2038	9	Reduce Index 9 but continue		
2038	x	2034	8	Reduce Index 8 and return to 2034		
2039	w	2025		Wait to 2025		
	k	2040				
	k	2041				
	k	2042				

RECOMP III TECHNICAL BULLETIN NO. 6

TITLE: RIP OUTPUT IN FIXED POINT FORMAT

PURPOSE: To describe a method whereby a RIP g function may allow the output of a result in Fixed Point Format.

EFFECTIVE DATE: 1 March 1962

CONTENTS:

1. Introduction
2. Programming the g function.
3. Steps to implement the g function.
4. Modifications of the Fixed Point Format.

REFERENCES:

R3S-10
R3S-11
R3P-16
RIP Programming Manual

AUTHOR: L. Laubscher

1. INTRODUCTION

In RIP, it is often desirable to output answers in fixed point format. There is no RIP command which will allow this to be done, but it may easily be implemented with a g function. Thus, in order to output a number in fixed point format, the RIP program would first have to pick up the number to be output and then perform the appropriate g function. For instance, the command sequence might be:

```
    a
g   8
```

Where a is the number to be output and 8 is the g function for fixed point output.

The particular fixed point format could be standard (such as one calling for 5 digits to be printed to the left of the decimal point and 4 digits to the right), or it may be able to be varied by the RIP program itself. (See Section 4.)

Sections 2 and 3 show how such a RIP g function could be made and how it would operate. The following assumes a familiarity with both RIP and machine language coding.

2. PROGRAMMING THE g FUNCTION

2. 1 Location of R3S-10:

The g function described above uses R3S-10 as an output subroutine. R3S-10 normally occupies 240_8 locations, but because it could be made to share coding with R3S-11 which is already located within RIP, the number of locations occupied by R3S-10 may be reduced to 170_8 locations or to 120 RIP addresses. In order to preserve as much contiguous memory in RIP as possible, this function will be located from location 7570_8 to location 7757_8 which corresponds to RIP locations 2880 to 2999 inclusive.

2. 2 Location and Length of the Program:

Two machine language instructions and one keyboard are necessary to perform this g function. Since one location is available in R3S-10 at 7731_8 or RIP address 2977, this will be used to hold the keyword. Location 7567_8 or RIP address 2879 will hold the program.

2.2 (Location and length of the program -- continued):

```
7567  CLA  KW       +37 7731.0
       TRA  R3S-10  +51 7570.0

7731   KW           +10 0504.0
                    +00 7763.1
```

2.3 Modification of RIP:

Since this function is to be g8, the following command will be inserted into location 1627 in accordance with Appendix I of the RIP Programming Manual.

```
1627   -75 6700.0
       -00 0330.0
```

2.4 Since R3S-10 saves the L loop, Index Register, and A Register, it is compatible with RIP, and the printout will be non-destructive, i.e., the contents of the RIP accumulator will not be changed by the g function.

3. STEPS TO IMPLEMENT THE g FUNCTION

3.1 Load RIP.

3.2 Load R3S-10 into location 7570 using the i function of Load/Start (R3P-1). The tape will not read in completely.

3.3 Return to Load/Start by pressing the RESET button. Using the c function of Load/Start, enter the following commands to modify R3S-10 such that it uses a table within RIP (Sec. 2.1).

```
7622   +16 0000.0
       +77 1101.0

7632   +16 0000.0
       +77 1077.0
```

3.4 Enter the program and keyword by using the c function of Load/Start.

```
7567   +37 7731.0
       +51 7570.0

7731   +10 0504.0
       +00 7763.1
```

3.5 Enter the g 8 function into RIP by placing the following command into 1627.

```
1627    -75 6700.0
        -00 0330.0
```

4. MODIFICATIONS OF THE FIXED POINT FORMAT

4.1 To type a space instead of a + (plus) sign, change:

RIP Address	Location	From	To
2978	7732	-010531.1 +340000.1	-010531.0 +340000.1

4.2 To type a lower case, ignoring the sign of the number, change:

RIP Address	Location	From	To
2892	7604	+010003.0 +160000.0	+010001.0 +160000.0

4.3 To eliminate the lower case, sign, and space, change:

RIP Address	Location	From	To
2891	7603	+537760.0 +777732.0	+560000.0 +517764.1

4.4 To change the number of digits typed out to the left and the right of the decimal point, set:

RIP Address	Location	To
2977	7731	+MMLLRR.0 +007763.1

Where MM is the number of significant digits to be printed in the fraction if floating point format is used. (Floating point format will be used if LL is too small.)

LL is the number of digits to the left of the decimal point if fixed point format is used, and

RR is the number of digits to the right of the decimal point if fixed point format is used.

4.5 The above changes may be made using Load/Start or by a RIP program which moves a previously stored keyword into address 2977. For example, suppose the keyword specifying 3 digits to the left and none to the right was stored in Location 7731 by using Load/Start:

RIP ADDRESS	Location	Keyword
2878	7566	+10 0300.0 +00 7763.1

Then the RIP command sequence:

```
  2878
s 2977
```

would cause all succeeding typeouts using g 8 to be in this format.

RECOMP III TECHNICAL BULLETIN NO. 10

TITLE: MACHINE LANGUAGE TO RIP CONVERSION TABLE

PURPOSE: When writing programs in machine language which are compatible with RIP, it is often necessary to convert octal addresses to RIP decimal addresses. To facilitate this conversion, a complete table of octal versus RIP addresses is published herewith.

EFFECTIVE DATE: 29 May 1962

CONTENTS: Conversion Table

AUTHOR: L. Laubscher

Machine Language To RIP Conversion Table

OCTAL	RIP	OCTAL	RIP	OCTAL	RIP	OCTAL	RIP	OCTAL	RIP
2070	0000	2140	0040	2210	0080	2260	0120	2330	0160
2071	0001	2141	0041	2211	0081	2261	0121	2331	0161
2072	0002	2142	0042	2212	0082	2262	0122	2332	0162
2073	0003	2143	0043	2213	0083	2263	0123	2333	0163
2074	0004	2144	0044	2214	0084	2264	0124	2334	0164
2075	0005	2145	0045	2215	0085	2265	0125	2335	0165
2076	0006	2146	0046	2216	0086	2266	0126	2336	0166
2077	0007	2147	0047	2217	0087	2267	0127	2337	0167
2100	0008	2150	0048	2220	0088	2270	0128	2340	0168
2101	0009	2151	0049	2221	0089	2271	0129	2341	0169
2102	0010	2152	0050	2222	0090	2272	0130	2342	0170
2103	0011	2153	0051	2223	0091	2273	0131	2343	0171
2104	0012	2154	0052	2224	0092	2274	0132	2344	0172
2105	0013	2155	0053	2225	0093	2275	0133	2345	0173
2106	0014	2156	0054	2226	0094	2276	0134	2346	0174
2107	0015	2157	0055	2227	0095	2277	0135	2347	0175
2110	0016	2160	0056	2230	0096	2300	0136	2350	0176
2111	0017	2161	0057	2231	0097	2301	0137	2351	0177
2112	0018	2162	0058	2232	0098	2302	0138	2352	0178
2113	0019	2163	0059	2233	0099	2303	0139	2353	0179
2114	0020	2164	0060	2234	0100	2304	0140	2354	0180
2115	0021	2165	0061	2235	0101	2305	0141	2355	0181
2116	0022	2166	0062	2236	0102	2306	0142	2356	0182
2117	0023	2167	0063	2237	0103	2307	0143	2357	0183
2120	0024	2170	0064	2240	0104	2310	0144	2360	0184
2121	0025	2171	0065	2241	0105	2311	0145	2361	0185
2122	0026	2172	0066	2242	0106	2312	0146	2362	0186
2123	0027	2173	0067	2243	0107	2313	0147	2363	0187
2124	0028	2174	0068	2244	0108	2314	0148	2364	0188
2125	0029	2175	0069	2245	0109	2315	0149	2365	0189
2126	0030	2176	0070	2246	0110	2316	0150	2366	0190
2127	0031	2177	0071	2247	0111	2317	0151	2367	0191
2130	0032	2200	0072	2250	0112	2320	0152	2370	0192
2131	0033	2201	0073	2251	0113	2321	0153	2371	0193
2132	0034	2202	0074	2252	0114	2322	0154	2372	0194
2133	0035	2203	0075	2253	0115	2323	0155	2373	0195
2134	0036	2204	0076	2254	0116	2324	0156	2374	0196
2135	0037	2205	0077	2255	0117	2325	0157	2375	0197
2136	0038	2206	0078	2256	0118	2326	0158	2376	0198
2137	0039	2207	0079	2257	0119	2327	0159	2377	0199

Machine Language To RIP Conversion Table

OCTAL	RIP	OCTAL	RIP	OCTAL	RIP	OCTAL	RIP	OCTAL	RIP
24ØØ	Ø2ØØ	245Ø	Ø24Ø	252Ø	Ø28Ø	257Ø	Ø32Ø	264Ø	Ø36Ø
24Ø1	Ø2Ø1	2451	Ø241	2521	Ø281	2571	Ø321	2641	Ø361
24Ø2	Ø2Ø2	2452	Ø242	2522	Ø282	2572	Ø322	2642	Ø362
24Ø3	Ø2Ø3	2453	Ø243	2523	Ø283	2573	Ø323	2643	Ø363
24Ø4	Ø2Ø4	2454	Ø244	2524	Ø284	2574	Ø324	2644	Ø364
24Ø5	Ø2Ø5	2455	Ø245	2525	Ø285	2575	Ø325	2645	Ø365
24Ø6	Ø2Ø6	2456	Ø246	2526	Ø286	2576	Ø326	2646	Ø366
24Ø7	Ø2Ø7	2457	Ø247	2527	Ø287	2577	Ø327	2647	Ø367
241Ø	Ø2Ø8	246Ø	Ø248	253Ø	Ø288	26ØØ	Ø328	265Ø	Ø368
2411	Ø2Ø9	2461	Ø249	2531	Ø289	26Ø1	Ø329	2651	Ø369
2412	Ø21Ø	2462	Ø25Ø	2532	Ø29Ø	26Ø2	Ø33Ø	2652	Ø37Ø
2413	Ø211	2463	Ø251	2533	Ø291	26Ø3	Ø331	2653	Ø371
2414	Ø212	2464	Ø252	2534	Ø292	26Ø4	Ø332	2654	Ø372
2415	Ø213	2465	Ø253	2535	Ø293	26Ø5	Ø333	2655	Ø373
2416	Ø214	2466	Ø254	2536	Ø294	26Ø6	Ø334	2656	Ø374
2417	Ø215	2467	Ø255	2537	Ø295	26Ø7	Ø335	2657	Ø375
242Ø	Ø216	247Ø	Ø256	254Ø	Ø296	261Ø	Ø336	266Ø	Ø376
2421	Ø217	2471	Ø257	2541	Ø297	2611	Ø337	2661	Ø377
2422	Ø218	2472	Ø258	2542	Ø298	2612	Ø338	2662	Ø378
2423	Ø219	2473	Ø259	2543	Ø299	2613	Ø339	2663	Ø379
2424	Ø22Ø	2474	Ø26Ø	2544	Ø3ØØ	2614	Ø34Ø	2664	Ø38Ø
2425	Ø221	2475	Ø261	2545	Ø3Ø1	2615	Ø341	2665	Ø381
2426	Ø222	2476	Ø262	2546	Ø3Ø2	2616	Ø342	2666	Ø382
2427	Ø223	2477	Ø263	2547	Ø3Ø3	2617	Ø343	2667	Ø383
243Ø	Ø224	25ØØ	Ø264	255Ø	Ø3Ø4	262Ø	Ø344	267Ø	Ø384
2431	Ø225	25Ø1	Ø265	2551	Ø3Ø5	2621	Ø345	2671	Ø385
2432	Ø226	25Ø2	Ø266	2552	Ø3Ø6	2622	Ø346	2672	Ø386
2433	Ø227	25Ø3	Ø267	2553	Ø3Ø7	2623	Ø347	2673	Ø387
2434	Ø228	25Ø4	Ø268	2554	Ø3Ø8	2624	Ø348	2674	Ø388
2435	Ø229	25Ø5	Ø269	2555	Ø3Ø9	2625	Ø349	2675	Ø389
2436	Ø23Ø	25Ø6	Ø27Ø	2556	Ø31Ø	2626	Ø35Ø	2676	Ø39Ø
2437	Ø231	25Ø7	Ø271	2557	Ø311	2627	Ø351	2677	Ø391
244Ø	Ø232	251Ø	Ø272	256Ø	Ø312	263Ø	Ø352	27ØØ	Ø392
2441	Ø233	2511	Ø273	2561	Ø313	2631	Ø353	27Ø1	Ø393
2442	Ø234	2512	Ø274	2562	Ø314	2632	Ø354	27Ø2	Ø394
2443	Ø235	2513	Ø275	2563	Ø315	2633	Ø355	27Ø3	Ø395
2444	Ø236	2514	Ø276	2564	Ø316	2634	Ø356	27Ø4	Ø396
2445	Ø237	2515	Ø277	2565	Ø317	2635	Ø357	27Ø5	Ø397
2446	Ø238	2516	Ø278	2566	Ø318	2636	Ø358	27Ø6	Ø398
2447	Ø239	2517	Ø279	2567	Ø319	2637	Ø359	27Ø7	Ø399

Machine Language To RIP Conversion Table

OCTAL RIP	OCTAL RIP	OCTAL RIP	OCTAL RIP	OCTAL RIP
271Ø---Ø4ØØ	276Ø---Ø44Ø	3Ø3Ø---Ø48Ø	31ØØ---Ø52Ø	315Ø---Ø56Ø
2711---Ø4Ø1	2761---Ø441	3Ø31---Ø481	31Ø1---Ø521	3151---Ø561
2712---Ø4Ø2	2762---Ø442	3Ø32---Ø482	31Ø2---Ø522	3152---Ø562
2713---Ø4Ø3	2763---Ø443	3Ø33---Ø483	31Ø3---Ø523	3153---Ø563
2714---Ø4Ø4	2764---Ø444	3Ø34---Ø484	31Ø4---Ø524	3154---Ø564
2715---Ø4Ø5	2765---Ø445	3Ø35---Ø485	31Ø5---Ø525	3155---Ø565
2716---Ø4Ø6	2766---Ø446	3Ø36---Ø486	31Ø6---Ø526	3156---Ø566
2717---Ø4Ø7	2767---Ø447	3Ø37---Ø487	31Ø7---Ø527	3157---Ø567
272Ø---Ø4Ø8	277Ø---Ø448	3Ø4Ø---Ø488	311Ø---Ø528	316Ø---Ø568
2721---Ø4Ø9	2771---Ø449	3Ø41---Ø489	3111---Ø529	3161---Ø569
2722---Ø41Ø	2772---Ø45Ø	3Ø42---Ø49Ø	3112---Ø53Ø	3162---Ø57Ø
2723---Ø411	2773---Ø451	3Ø43---Ø491	3113---Ø531	3163---Ø571
2724---Ø412	2774---Ø452	3Ø44---Ø492	3114---Ø532	3164---Ø572
2725---Ø413	2775---Ø453	3Ø45---Ø493	3115---Ø533	3165---Ø573
2726---Ø414	2776---Ø454	3Ø46---Ø494	3116---Ø534	3166---Ø574
2727---Ø415	2777---Ø455	3Ø47---Ø495	3117---Ø535	3167---Ø575
273Ø---Ø416	3ØØØ---Ø456	3Ø5Ø---Ø496	312Ø---Ø536	317Ø---Ø576
2731---Ø417	3ØØ1---Ø457	3Ø51---Ø497	3121---Ø537	3171---Ø577
2732---Ø418	3ØØ2---Ø458	3Ø52---Ø498	3122---Ø538	3172---Ø578
2733---Ø419	3ØØ3---Ø459	3Ø53---Ø499	3123---Ø539	3173---Ø579
2734---Ø42Ø	3ØØ4---Ø46Ø	3Ø54---Ø5ØØ	3124---Ø54Ø	3174---Ø58Ø
2735---Ø421	3ØØ5---Ø461	3Ø55---Ø5Ø1	3125---Ø541	3175---Ø581
2736---Ø422	3ØØ6---Ø462	3Ø56---Ø5Ø2	3126---Ø542	3176---Ø582
2737---Ø423	3ØØ7---Ø463	3Ø57---Ø5Ø3	3127---Ø543	3177---Ø583
274Ø---Ø424	3Ø1Ø---Ø464	3Ø6Ø---Ø5Ø4	313Ø---Ø544	32ØØ---Ø584
2741---Ø425	3Ø11---Ø465	3Ø61---Ø5Ø5	3131---Ø545	32Ø1---Ø585
2742---Ø426	3Ø12---Ø466	3Ø62---Ø5Ø6	3132---Ø546	32Ø2---Ø586
2743---Ø427	3Ø13---Ø467	3Ø63---Ø5Ø7	3133---Ø547	32Ø3---Ø587
2744---Ø428	3Ø14---Ø468	3Ø64---Ø5Ø8	3134---Ø548	32Ø4---Ø588
2745---Ø429	3Ø15---Ø469	3Ø65---Ø5Ø9	3135---Ø549	32Ø5---Ø589
2746---Ø43Ø	3Ø16---Ø47Ø	3Ø66---Ø51Ø	3136---Ø55Ø	32Ø6---Ø59Ø
2747---Ø431	3Ø17---Ø471	3Ø67---Ø511	3137---Ø551	32Ø7---Ø591
275Ø---Ø432	3Ø2Ø---Ø472	3Ø7Ø---Ø512	314Ø---Ø552	321Ø---Ø592
2751---Ø433	3Ø21---Ø473	3Ø71---Ø513	3141---Ø553	3211---Ø593
2752---Ø434	3Ø22---Ø474	3Ø72---Ø514	3142---Ø554	3212---Ø594
2753---Ø435	3Ø23---Ø475	3Ø73---Ø515	3143---Ø555	3213---Ø595
2754---Ø436	3Ø24---Ø476	3Ø74---Ø516	3144---Ø556	3214---Ø596
2755---Ø437	3Ø25---Ø477	3Ø75---Ø517	3145---Ø557	3215---Ø597
2756---Ø438	3Ø26---Ø478	3Ø76---Ø518	3146---Ø558	3216---Ø598
2757---Ø439	3Ø27---Ø479	3Ø77---Ø519	3147---Ø559	3217---Ø599

Machine Language To RIP Conversion Table

OCTAL	RIP	OCTAL	RIP	OCTAL	RIP	OCTAL	RIP	OCTAL	RIP
3220	0600	3270	0640	3340	0680	3410	0720	3460	0760
3221	0601	3271	0641	3341	0681	3411	0721	3461	0761
3222	0602	3272	0642	3342	0682	3412	0722	3462	0762
3223	0603	3273	0643	3343	0683	3413	0723	3463	0763
3224	0604	3274	0644	3344	0684	3414	0724	3464	0764
3225	0605	3275	0645	3345	0685	3415	0725	3465	0765
3226	0606	3276	0646	3346	0686	3416	0726	3466	0766
3227	0607	3277	0647	3347	0687	3417	0727	3467	0767
3230	0608	3300	0648	3350	0688	3420	0728	3470	0768
3231	0609	3301	0649	3351	0689	3421	0729	3471	0769
3232	0610	3302	0650	3352	0690	3422	0730	3472	0770
3233	0611	3303	0651	3353	0691	3423	0731	3473	0771
3234	0612	3304	0652	3354	0692	3424	0732	3474	0772
3235	0613	3305	0653	3355	0693	3425	0733	3475	0773
3236	0614	3306	0654	3356	0694	3426	0734	3476	0774
3237	0615	3307	0655	3357	0695	3427	0735	3477	0775
3240	0616	3310	0656	3360	0696	3430	0736	3500	0776
3241	0617	3311	0657	3361	0697	3431	0737	3501	0777
3242	0618	3312	0658	3362	0698	3432	0738	3502	0778
3243	0619	3313	0659	3363	0699	3433	0739	3503	0779
3244	0620	3314	0660	3364	0700	3434	0740	3504	0780
3245	0621	3315	0661	3365	0701	3435	0741	3505	0781
3246	0622	3316	0662	3366	0702	3436	0742	3506	0782
3247	0623	3317	0663	3367	0703	3437	0743	3507	0783
3250	0624	3320	0664	3370	0704	3440	0744	3510	0784
3251	0625	3321	0665	3371	0705	3441	0745	3511	0785
3252	0626	3322	0666	3372	0706	3442	0746	3512	0786
3253	0627	3323	0667	3373	0707	3443	0747	3513	0787
3254	0628	3324	0668	3374	0708	3444	0748	3514	0788
3255	0629	3325	0669	3375	0709	3445	0749	3515	0789
3256	0630	3326	0670	3376	0710	3446	0750	3516	0790
3257	0631	3327	0671	3377	0711	3447	0751	3517	0791
3260	0632	3330	0672	3400	0712	3450	0752	3520	0792
3261	0633	3331	0673	3401	0713	3451	0753	3521	0793
3262	0634	3332	0674	3402	0714	3452	0754	3522	0794
3263	0635	3333	0675	3403	0715	3453	0755	3523	0795
3264	0636	3334	0676	3404	0716	3454	0756	3524	0796
3265	0637	3335	0677	3405	0717	3455	0757	3525	0797
3266	0638	3336	0678	3406	0718	3456	0758	3526	0798
3267	0639	3337	0679	3407	0719	3457	0759	3527	0799

Machine Language To RIP Conversion Table

OCTAL	RIP	OCTAL	RIP	OCTAL	RIP	OCTAL	RIP	OCTAL	RIP
353Ø	Ø8ØØ	36ØØ	Ø84Ø	365Ø	Ø88Ø	372Ø	Ø92Ø	377Ø	Ø96Ø
3531	Ø8Ø1	36Ø1	Ø841	3651	Ø881	3721	Ø921	3771	Ø961
3532	Ø8Ø2	36Ø2	Ø842	3652	Ø882	3722	Ø922	3772	Ø962
3533	Ø8Ø3	36Ø3	Ø843	3653	Ø883	3723	Ø923	3773	Ø963
3534	Ø8Ø4	36Ø4	Ø844	3654	Ø884	3724	Ø924	3774	Ø964
3535	Ø8Ø5	36Ø5	Ø845	3655	Ø885	3725	Ø925	3775	Ø965
3536	Ø8Ø6	36Ø6	Ø846	3656	Ø886	3726	Ø926	3776	Ø966
3537	Ø8Ø7	36Ø7	Ø847	3657	Ø887	3727	Ø927	3777	Ø967
354Ø	Ø8Ø8	361Ø	Ø848	366Ø	Ø888	373Ø	Ø928	4ØØØ	Ø968
3541	Ø8Ø9	3611	Ø849	3661	Ø889	3731	Ø929	4ØØ1	Ø969
3542	Ø81Ø	3612	Ø85Ø	3662	Ø89Ø	3732	Ø93Ø	4ØØ2	Ø97Ø
3543	Ø811	3613	Ø851	3663	Ø891	3733	Ø931	4ØØ3	Ø971
3544	Ø812	3614	Ø852	3664	Ø892	3734	Ø932	4ØØ4	Ø972
3545	Ø813	3615	Ø853	3665	Ø893	3735	Ø933	4ØØ5	Ø973
3546	Ø814	3616	Ø854	3666	Ø894	3736	Ø934	4ØØ6	Ø974
3547	Ø815	3617	Ø855	3667	Ø895	3737	Ø935	4ØØ7	Ø975
355Ø	Ø816	362Ø	Ø856	367Ø	Ø896	374Ø	Ø936	4Ø1Ø	Ø976
3551	Ø817	3621	Ø857	3671	Ø897	3741	Ø937	4Ø11	Ø977
3552	Ø818	3622	Ø858	3672	Ø898	3742	Ø938	4Ø12	Ø978
3553	Ø819	3623	Ø859	3673	Ø899	3743	Ø939	4Ø13	Ø979
3554	Ø82Ø	3624	Ø86Ø	3674	Ø9ØØ	3744	Ø94Ø	4Ø14	Ø98Ø
3555	Ø821	3625	Ø861	3675	Ø9Ø1	3745	Ø941	4Ø15	Ø981
3556	Ø822	3626	Ø862	3676	Ø9Ø2	3746	Ø942	4Ø16	Ø982
3557	Ø823	3627	Ø863	3677	Ø9Ø3	3747	Ø943	4Ø17	Ø983
356Ø	Ø824	363Ø	Ø864	37ØØ	Ø9Ø4	375Ø	Ø944	4Ø2Ø	Ø984
3561	Ø825	3631	Ø865	37Ø1	Ø9Ø5	3751	Ø945	4Ø21	Ø985
3562	Ø826	3632	Ø866	37Ø2	Ø9Ø6	3752	Ø946	4Ø22	Ø986
3563	Ø827	3633	Ø867	37Ø3	Ø9Ø7	3753	Ø947	4Ø23	Ø987
3564	Ø828	3634	Ø868	37Ø4	Ø9Ø8	3754	Ø948	4Ø24	Ø988
3565	Ø829	3635	Ø869	37Ø5	Ø9Ø9	3755	Ø949	4Ø25	Ø989
3566	Ø83Ø	3636	Ø87Ø	37Ø6	Ø91Ø	3756	Ø95Ø	4Ø26	Ø99Ø
3567	Ø831	3637	Ø871	37Ø7	Ø911	3757	Ø951	4Ø27	Ø991
357Ø	Ø832	364Ø	Ø872	371Ø	Ø912	376Ø	Ø952	4Ø3Ø	Ø992
3571	Ø833	3641	Ø873	3711	Ø913	3761	Ø953	4Ø31	Ø993
3572	Ø834	3642	Ø874	3712	Ø914	3762	Ø954	4Ø32	Ø994
3573	Ø835	3643	Ø875	3713	Ø915	3763	Ø955	4Ø33	Ø995
3574	Ø836	3644	Ø876	3714	Ø916	3764	Ø956	4Ø34	Ø996
3575	Ø837	3645	Ø877	3715	Ø917	3765	Ø957	4Ø35	Ø997
3576	Ø838	3646	Ø878	3716	Ø918	3766	Ø958	4Ø36	Ø998
3577	Ø839	3647	Ø879	3717	Ø919	3767	Ø959	4Ø37	Ø999

Machine Language To RIP Conversion Table

OCTAL	RIP	OCTAL	RIP	OCTAL	RIP	OCTAL	RIP	OCTAL	RIP
4Ø4Ø	1ØØØ	411Ø	1Ø4Ø	416Ø	1Ø8Ø	423Ø	112Ø	43ØØ	116Ø
4Ø41	1ØØ1	4111	1Ø41	4161	1Ø81	4231	1121	43Ø1	1161
4Ø42	1ØØ2	4112	1Ø42	4162	1Ø82	4232	1122	43Ø2	1162
4Ø43	1ØØ3	4113	1Ø43	4163	1Ø83	4233	1123	43Ø3	1163
4Ø44	1ØØ4	4114	1Ø44	4164	1Ø84	4234	1124	43Ø4	1164
4Ø45	1ØØ5	4115	1Ø45	4165	1Ø85	4235	1125	43Ø5	1165
4Ø46	1ØØ6	4116	1Ø46	4166	1Ø86	4236	1126	43Ø6	1166
4Ø47	1ØØ7	4117	1Ø47	4167	1Ø87	4237	1127	43Ø7	1167
4Ø5Ø	1ØØ8	412Ø	1Ø48	417Ø	1Ø88	424Ø	1128	431Ø	1168
4Ø51	1ØØ9	4121	1Ø49	4171	1Ø89	4241	1129	4311	1169
4Ø52	1Ø1Ø	4122	1Ø5Ø	4172	1Ø9Ø	4242	113Ø	4312	117Ø
4Ø53	1Ø11	4123	1Ø51	4173	1Ø91	4243	1131	4313	1171
4Ø54	1Ø12	4124	1Ø52	4174	1Ø92	4244	1132	4314	1172
4Ø55	1Ø13	4125	1Ø53	4175	1Ø93	4245	1133	4315	1173
4Ø56	1Ø14	4126	1Ø54	4176	1Ø94	4246	1134	4316	1174
4Ø57	1Ø15	4127	1Ø55	4177	1Ø95	4247	1135	4317	1175
4Ø6Ø	1Ø16	413Ø	1Ø56	42ØØ	1Ø96	425Ø	1136	432Ø	1176
4Ø61	1Ø17	4131	1Ø57	42Ø1	1Ø97	4251	1137	4321	1177
4Ø62	1Ø18	4132	1Ø58	42Ø2	1Ø98	4252	1138	4322	1178
4Ø63	1Ø19	4133	1Ø59	42Ø3	1Ø99	4253	1139	4323	1179
4Ø64	1Ø2Ø	4134	1Ø6Ø	42Ø4	11ØØ	4254	114Ø	4324	118Ø
4Ø65	1Ø21	4135	1Ø61	42Ø5	11Ø1	4255	1141	4325	1181
4Ø66	1Ø22	4136	1Ø62	42Ø6	11Ø2	4256	1142	4326	1182
4Ø67	1Ø23	4137	1Ø63	42Ø7	11Ø3	4257	1143	4327	1183
4Ø7Ø	1Ø24	414Ø	1Ø64	421Ø	11Ø4	426Ø	1144	433Ø	1184
4Ø71	1Ø25	4141	1Ø65	4211	11Ø5	4261	1145	4331	1185
4Ø72	1Ø26	4142	1Ø66	4212	11Ø6	4262	1146	4332	1186
4Ø73	1Ø27	4143	1Ø67	4213	11Ø7	4263	1147	4333	1187
4Ø74	1Ø28	4144	1Ø68	4214	11Ø8	4264	1148	4334	1188
4Ø75	1Ø29	4145	1Ø69	4215	11Ø9	4265	1149	4335	1189
4Ø76	1Ø3Ø	4146	1Ø7Ø	4216	111Ø	4266	115Ø	4336	119Ø
4Ø77	1Ø31	4147	1Ø71	4217	1111	4267	1151	4337	1191
41ØØ	1Ø32	415Ø	1Ø72	422Ø	1112	427Ø	1152	434Ø	1192
41Ø1	1Ø33	4151	1Ø73	4221	1113	4271	1153	4341	1193
41Ø2	1Ø34	4152	1Ø74	4222	1114	4272	1154	4342	1194
41Ø3	1Ø35	4153	1Ø75	4223	1115	4273	1155	4343	1195
41Ø4	1Ø36	4154	1Ø76	4224	1116	4274	1156	4344	1196
41Ø5	1Ø37	4155	1Ø77	4225	1117	4275	1157	4345	1197
41Ø6	1Ø38	4156	1Ø78	4226	1118	4276	1158	4346	1198
41Ø7	1Ø39	4157	1Ø79	4227	1119	4277	1159	4347	1199

Machine Language To RIP Conversion Table

OCTAL	RIP	OCTAL	RIP	OCTAL	RIP	OCTAL	RIP	OCTAL	RIP
435Ø	12ØØ	442Ø	124Ø	447Ø	128Ø	454Ø	132Ø	461Ø	136Ø
4351	12Ø1	4421	1241	4471	1281	4541	1321	4611	1361
4352	12Ø2	4422	1242	4472	1282	4542	1322	4612	1362
4353	12Ø3	4423	1243	4473	1283	4543	1323	4613	1363
4354	12Ø4	4424	1244	4474	1284	4544	1324	4614	1364
4355	12Ø5	4425	1245	4475	1285	4545	1325	4615	1365
4356	12Ø6	4426	1246	4476	1286	4546	1326	4616	1366
4357	12Ø7	4427	1247	4477	1287	4547	1327	4617	1367
436Ø	12Ø8	443Ø	1248	45ØØ	1288	455Ø	1328	462Ø	1368
4361	12Ø9	4431	1249	45Ø1	1289	4551	1329	4621	1369
4362	121Ø	4432	125Ø	45Ø2	129Ø	4552	133Ø	4622	137Ø
4363	1211	4433	1251	45Ø3	1291	4553	1331	4623	1371
4364	1212	4434	1252	45Ø4	1292	4554	1332	4624	1372
4365	1213	4435	1253	45Ø5	1293	4555	1333	4625	1373
4366	1214	4436	1254	45Ø6	1294	4556	1334	4626	1374
4367	1215	4437	1255	45Ø7	1295	4557	1335	4627	1375
437Ø	1216	444Ø	1256	451Ø	1296	456Ø	1336	463Ø	1376
4371	1217	4441	1257	4511	1297	4561	1337	4631	1377
4372	1218	4442	1258	4512	1298	4562	1338	4632	1378
4373	1219	4443	1259	4513	1299	4563	1339	4633	1379
4374	122Ø	4444	126Ø	4514	13ØØ	4564	134Ø	4634	138Ø
4375	1221	4445	1261	4515	13Ø1	4565	1341	4635	1381
4376	1222	4446	1262	4516	13Ø2	4566	1342	4636	1382
4377	1223	4447	1263	4517	13Ø3	4567	1343	4637	1383
44ØØ	1224	445Ø	1264	452Ø	13Ø4	457Ø	1344	464Ø	1384
44Ø1	1225	4451	1265	4521	13Ø5	4571	1345	4641	1385
44Ø2	1226	4452	1266	4522	13Ø6	4572	1346	4642	1386
44Ø3	1227	4453	1267	4523	13Ø7	4573	1347	4643	1387
44Ø4	1228	4454	1268	4524	13Ø8	4574	1348	4644	1388
44Ø5	1229	4455	1269	4525	13Ø9	4575	1349	4645	1389
44Ø6	123Ø	4456	127Ø	4526	131Ø	4576	135Ø	4646	139Ø
44Ø7	1231	4457	1271	4527	1311	4577	1351	4647	1391
441Ø	1232	446Ø	1272	453Ø	1312	46ØØ	1352	465Ø	1392
4411	1233	4461	1273	4531	1313	46Ø1	1353	4651	1393
4412	1234	4462	1274	4532	1314	46Ø2	1354	4652	1394
4413	1235	4463	1275	4533	1315	46Ø3	1355	4653	1395
4414	1236	4464	1276	4534	1316	46Ø4	1356	4654	1396
4415	1237	4465	1277	4535	1317	46Ø5	1357	4655	1397
4416	1238	4466	1278	4536	1318	46Ø6	1358	4656	1398
4417	1239	4467	1279	4537	1319	46Ø7	1359	4657	1399

Machine Language To RIP Conversion Table

OCTAL	RIP	OCTAL	RIP	OCTAL	RIP	OCTAL	RIP	OCTAL	RIP
4660	1400	4730	1440	5000	1480	5050	1520	5120	1560
4661	1401	4731	1441	5001	1481	5051	1521	5121	1561
4662	1402	4732	1442	5002	1482	5052	1522	5122	1562
4663	1403	4733	1443	5003	1483	5053	1523	5123	1563
4664	1404	4734	1444	5004	1484	5054	1524	5124	1564
4665	1405	4735	1445	5005	1485	5055	1525	5125	1565
4666	1406	4736	1446	5006	1486	5056	1526	5126	1566
4667	1407	4737	1447	5007	1487	5057	1527	5127	1567
4670	1408	4740	1448	5010	1488	5060	1528	5130	1568
4671	1409	4741	1449	5011	1489	5061	1529	5131	1569
4672	1410	4742	1450	5012	1490	5062	1530	5132	1570
4673	1411	4743	1451	5013	1491	5063	1531	5133	1571
4674	1412	4744	1452	5014	1492	5064	1532	5134	1572
4675	1413	4745	1453	5015	1493	5065	1533	5135	1573
4676	1414	4746	1454	5016	1494	5066	1534	5136	1574
4677	1415	4747	1455	5017	1495	5067	1535	5137	1575
4700	1416	4750	1456	5020	1496	5070	1536	5140	1576
4701	1417	4751	1457	5021	1497	5071	1537	5141	1577
4702	1418	4752	1458	5022	1498	5072	1538	5142	1578
4703	1419	4753	1459	5023	1499	5073	1539	5143	1579
4704	1420	4754	1460	5024	1500	5074	1540	5144	1580
4705	1421	4755	1461	5025	1501	5075	1541	5145	1581
4706	1422	4756	1462	5026	1502	5076	1542	5146	1582
4707	1423	4757	1463	5027	1503	5077	1543	5147	1583
4710	1424	4760	1464	5030	1504	5100	1544	5150	1584
4711	1425	4761	1465	5031	1505	5101	1545	5151	1585
4712	1426	4762	1466	5032	1506	5102	1546	5152	1586
4713	1427	4763	1467	5033	1507	5103	1547	5153	1587
4714	1428	4764	1468	5034	1508	5104	1548	5154	1588
4715	1429	4765	1469	5035	1509	5105	1549	5155	1589
4716	1430	4766	1470	5036	1510	5106	1550	5156	1590
4717	1431	4767	1471	5037	1511	5107	1551	5157	1591
4720	1432	4770	1472	5040	1512	5110	1552	5160	1592
4721	1433	4771	1473	5041	1513	5111	1553	5161	1593
4722	1434	4772	1474	5042	1514	5112	1554	5162	1594
4723	1435	4773	1475	5043	1515	5113	1555	5163	1595
4724	1436	4774	1476	5044	1516	5114	1556	5164	1596
4725	1437	4775	1477	5045	1517	5115	1557	5165	1597
4726	1438	4776	1478	5046	1518	5116	1558	5166	1598
4727	1439	4777	1479	5047	1519	5117	1559	5167	1599

Machine Language To RIP Conversion Table

OCTAL RIP	OCTAL RIP	OCTAL RIP	OCTAL RIP	OCTAL RIP
5170---1600	5240---1640	5310---1680	5360---1720	5430---1760
5171---1601	5241---1641	5311---1681	5361---1721	5431---1761
5172---1602	5242---1642	5312---1682	5362---1722	5432---1762
5173---1603	5243---1643	5313---1683	5363---1723	5433---1763
5174---1604	5244---1644	5314---1684	5364---1724	5434---1764
5175---1605	5245---1645	5315---1685	5365---1725	5435---1765
5176---1606	5246---1646	5316---1686	5366---1726	5436---1766
5177---1607	5247---1647	5317---1687	5367---1727	5437---1767
5200---1608	5250---1648	5320---1688	5370---1728	5440---1768
5201---1609	5251---1649	5321---1689	5371---1729	5441---1769
5202---1610	5252---1650	5322---1690	5372---1730	5442---1770
5203---1611	5253---1651	5323---1691	5373---1731	5443---1771
5204---1612	5254---1652	5324---1692	5374---1732	5444---1772
5205---1613	5255---1653	5325---1693	5375---1733	5445---1773
5206---1614	5256---1654	5326---1694	5376---1734	5446---1774
5207---1615	5257---1655	5327---1695	5377---1735	5447---1775
5210---1616	5260---1656	5330---1696	5400---1736	5450---1776
5211---1617	5261---1657	5331---1697	5401---1737	5451---1777
5212---1618	5262---1658	5332---1698	5402---1738	5452---1778
5213---1619	5263---1659	5333---1699	5403---1739	5453---1779
5214---1620	5264---1660	5334---1700	5404---1740	5454---1780
5215---1621	5265---1661	5335---1701	5405---1741	5455---1781
5216---1622	5266---1662	5336---1702	5406---1742	5456---1782
5217---1623	5267---1663	5337---1703	5407---1743	5457---1783
5220---1624	5270---1664	5340---1704	5410---1744	5460---1784
5221---1625	5271---1665	5341---1705	5411---1745	5461---1785
5222---1626	5272---1666	5342---1706	5412---1746	5462---1786
5223---1627	5273---1667	5343---1707	5413---1747	5463---1787
5224---1628	5274---1668	5344---1708	5414---1748	5464---1788
5225---1629	5275---1669	5345---1709	5415---1749	5465---1789
5226---1630	5276---1670	5346---1710	5416---1750	5466---1790
5227---1631	5277---1671	5347---1711	5417---1751	5467---1791
5230---1632	5300---1672	5350---1712	5420---1752	5470---1792
5231---1633	5301---1673	5351---1713	5421---1753	5471---1793
5232---1634	5302---1674	5352---1714	5422---1754	5472---1794
5233---1635	5303---1675	5353---1715	5423---1755	5473---1795
5234---1636	5304---1676	5354---1716	5424---1756	5474---1796
5235---1637	5305---1677	5355---1717	5425---1757	5475---1797
5236---1638	5306---1678	5356---1718	5426---1758	5476---1798
5237---1639	5307---1679	5357---1719	5427---1759	5477---1799

Machine Language To RIP Conversion Table

OCTAL	RIP	OCTAL	RIP	OCTAL	RIP	OCTAL	RIP	OCTAL	RIP
55ØØ	18ØØ	555Ø	184Ø	562Ø	188Ø	567Ø	192Ø	574Ø	196Ø
55Ø1	18Ø1	5551	1841	5621	1881	5671	1921	5741	1961
55Ø2	18Ø2	5552	1842	5622	1882	5672	1922	5742	1962
55Ø3	18Ø3	5553	1843	5623	1883	5673	1923	5743	1963
55Ø4	18Ø4	5554	1844	5624	1884	5674	1924	5744	1964
55Ø5	18Ø5	5555	1845	5625	1885	5675	1925	5745	1965
55Ø6	18Ø6	5556	1846	5626	1886	5676	1926	5746	1966
55Ø7	18Ø7	5557	1847	5627	1887	5677	1927	5747	1967
551Ø	18Ø8	556Ø	1848	563Ø	1888	57ØØ	1928	575Ø	1968
5511	18Ø9	5561	1849	5631	1889	57Ø1	1929	5751	1969
5512	181Ø	5562	185Ø	5632	189Ø	57Ø2	193Ø	5752	197Ø
5513	1811	5563	1851	5633	1891	57Ø3	1931	5753	1971
5514	1812	5564	1852	5634	1892	57Ø4	1932	5754	1972
5515	1813	5565	1853	5635	1893	57Ø5	1933	5755	1973
5516	1814	5566	1854	5636	1894	57Ø6	1934	5756	1974
5517	1815	5567	1855	5637	1895	57Ø7	1935	5757	1975
552Ø	1816	557Ø	1856	564Ø	1896	571Ø	1936	576Ø	1976
5521	1817	5571	1857	5641	1897	5711	1937	5761	1977
5522	1818	5572	1858	5642	1898	5712	1938	5762	1978
5523	1819	5573	1859	5643	1899	5713	1939	5763	1979
5524	182Ø	5574	186Ø	5644	19ØØ	5714	194Ø	5764	198Ø
5525	1821	5575	1861	5645	19Ø1	5715	1941	5765	1981
5526	1822	5576	1862	5646	19Ø2	5716	1942	5766	1982
5527	1823	5577	1863	5647	19Ø3	5717	1943	5767	1983
553Ø	1824	56ØØ	1864	565Ø	19Ø4	572Ø	1944	577Ø	1984
5531	1825	56Ø1	1865	5651	19Ø5	5721	1945	5771	1985
5532	1826	56Ø2	1866	5652	19Ø6	5722	1946	5772	1986
5533	1827	56Ø3	1867	5653	19Ø7	5723	1947	5773	1987
5534	1828	56Ø4	1868	5654	19Ø8	5724	1948	5774	1988
5535	1829	56Ø5	1869	5655	19Ø9	5725	1949	5775	1989
5536	183Ø	56Ø6	187Ø	5656	191Ø	5726	195Ø	5776	199Ø
5537	1831	56Ø7	1871	5657	1911	5727	1951	5777	1991
554Ø	1832	561Ø	1872	566Ø	1912	573Ø	1952	6ØØØ	1992
5541	1833	5611	1873	5661	1913	5731	1953	6ØØ1	1993
5542	1834	5612	1874	5662	1914	5732	1954	6ØØ2	1994
5543	1835	5613	1875	5663	1915	5733	1955	6ØØ3	1995
5544	1836	5614	1876	5664	1916	5734	1956	6ØØ4	1996
5545	1837	5615	1877	5665	1917	5735	1957	6ØØ5	1997
5546	1838	5616	1878	5666	1918	5736	1958	6ØØ6	1998
5547	1839	5617	1879	5667	1919	5737	1959	6ØØ7	1999

Machine Language To RIP Conversion Table

OCTAL	RIP	OCTAL	RIP	OCTAL	RIP	OCTAL	RIP	OCTAL	RIP
6Ø1Ø	2ØØØ	6Ø6Ø	2Ø4Ø	613Ø	2Ø8Ø	62ØØ	212Ø	625Ø	216Ø
6Ø11	2ØØ1	6Ø61	2Ø41	6131	2Ø81	62Ø1	2121	6251	2161
6Ø12	2ØØ2	6Ø62	2Ø42	6132	2Ø82	62Ø2	2122	6252	2162
6Ø13	2ØØ3	6Ø63	2Ø43	6133	2Ø83	62Ø3	2123	6253	2163
6Ø14	2ØØ4	6Ø64	2Ø44	6134	2Ø84	62Ø4	2124	6254	2164
6Ø15	2ØØ5	6Ø65	2Ø45	6135	2Ø85	62Ø5	2125	6255	2165
6Ø16	2ØØ6	6Ø66	2Ø46	6136	2Ø86	62Ø6	2126	6256	2166
6Ø17	2ØØ7	6Ø67	2Ø47	6137	2Ø87	62Ø7	2127	6257	2167
6Ø2Ø	2ØØ8	6Ø7Ø	2Ø48	614Ø	2Ø88	621Ø	2128	626Ø	2168
6Ø21	2ØØ9	6Ø71	2Ø49	6141	2Ø89	6211	2129	6261	2169
6Ø22	2Ø1Ø	6Ø72	2Ø5Ø	6142	2Ø9Ø	6212	213Ø	6262	217Ø
6Ø23	2Ø11	6Ø73	2Ø51	6143	2Ø91	6213	2131	6263	2171
6Ø24	2Ø12	6Ø74	2Ø52	6144	2Ø92	6214	2132	6264	2172
6Ø25	2Ø13	6Ø75	2Ø53	6145	2Ø93	6215	2133	6265	2173
6Ø26	2Ø14	6Ø76	2Ø54	6146	2Ø94	6216	2134	6266	2174
6Ø27	2Ø15	6Ø77	2Ø55	6147	2Ø95	6217	2135	6267	2175
6Ø3Ø	2Ø16	61ØØ	2Ø56	615Ø	2Ø96	622Ø	2136	627Ø	2176
6Ø31	2Ø17	61Ø1	2Ø57	6151	2Ø97	6221	2137	6271	2177
6Ø32	2Ø18	61Ø2	2Ø58	6152	2Ø98	6222	2138	6272	2178
6Ø33	2Ø19	61Ø3	2Ø59	6153	2Ø99	6223	2139	6273	2179
6Ø34	2Ø2Ø	61Ø4	2Ø6Ø	6154	21ØØ	6224	214Ø	6274	218Ø
6Ø35	2Ø21	61Ø5	2Ø61	6155	21Ø1	6225	2141	6275	2181
6Ø36	2Ø22	61Ø6	2Ø62	6156	21Ø2	6226	2142	6276	2182
6Ø37	2Ø23	61Ø7	2Ø63	6157	21Ø3	6227	2143	6277	2183
6Ø4Ø	2Ø24	611Ø	2Ø64	616Ø	21Ø4	623Ø	2144	63ØØ	2184
6Ø41	2Ø25	6111	2Ø65	6161	21Ø5	6231	2145	63Ø1	2185
6Ø42	2Ø26	6112	2Ø66	6162	21Ø6	6232	2146	63Ø2	2186
6Ø43	2Ø27	6113	2Ø67	6163	21Ø7	6233	2147	63Ø3	2187
6Ø44	2Ø28	6114	2Ø68	6164	21Ø8	6234	2148	63Ø4	2188
6Ø45	2Ø29	6115	2Ø69	6165	21Ø9	6235	2149	63Ø5	2189
6Ø46	2Ø3Ø	6116	2Ø7Ø	6166	211Ø	6236	215Ø	63Ø6	219Ø
6Ø47	2Ø31	6117	2Ø71	6167	2111	6237	2151	63Ø7	2191
6Ø5Ø	2Ø32	612Ø	2Ø72	617Ø	2112	624Ø	2152	631Ø	2192
6Ø51	2Ø33	6121	2Ø73	6171	2113	6241	2153	6311	2193
6Ø52	2Ø34	6122	2Ø74	6172	2114	6242	2154	6312	2194
6Ø53	2Ø35	6123	2Ø75	6173	2115	6243	2155	6313	2195
6Ø54	2Ø36	6124	2Ø76	6174	2116	6244	2156	6314	2196
6Ø55	2Ø37	6125	2Ø77	6175	2117	6245	2157	6315	2197
6Ø56	2Ø38	6126	2Ø78	6176	2118	6246	2158	6316	2198
6Ø57	2Ø39	6127	2Ø79	6177	2119	6247	2159	6317	2199

17

Machine Language To RIP Conversion Table

OCTAL	RIP	OCTAL	RIP	OCTAL	RIP	OCTAL	RIP	OCTAL	RIP
632Ø	22ØØ	637Ø	224Ø	644Ø	228Ø	651Ø	232Ø	656Ø	236Ø
6321	22Ø1	6371	2241	6441	2281	6511	2321	6561	2361
6322	22Ø2	6372	2242	6442	2282	6512	2322	6562	2362
6323	22Ø3	6373	2243	6443	2283	6513	2323	6563	2363
6324	22Ø4	6374	2244	6444	2284	6514	2324	6564	2364
6325	22Ø5	6375	2245	6445	2285	6515	2325	6565	2365
6326	22Ø6	6376	2246	6446	2286	6516	2326	6566	2366
6327	22Ø7	6377	2247	6447	2287	6517	2327	6567	2367
633Ø	22Ø8	64ØØ	2248	645Ø	2288	652Ø	2328	657Ø	2368
6331	22Ø9	64Ø1	2249	6451	2289	6521	2329	6571	2369
6332	221Ø	64Ø2	225Ø	6452	229Ø	6522	233Ø	6572	237Ø
6333	2211	64Ø3	2251	6453	2291	6523	2331	6573	2371
6334	2212	64Ø4	2252	6454	2292	6524	2332	6574	2372
6335	2213	64Ø5	2253	6455	2293	6525	2333	6575	2373
6336	2214	64Ø6	2254	6456	2294	6526	2334	6576	2374
6337	2215	64Ø7	2255	6457	2295	6527	2335	6577	2375
634Ø	2216	641Ø	2256	646Ø	2296	653Ø	2336	66ØØ	2376
6341	2217	6411	2257	6461	2297	6531	2337	66Ø1	2377
6342	2218	6412	2258	6462	2298	6532	2338	66Ø2	2378
6343	2219	6413	2259	6463	2299	6533	2339	66Ø3	2379
6344	222Ø	6414	226Ø	6464	23ØØ	6534	234Ø	66Ø4	238Ø
6345	2221	6415	2261	6465	23Ø1	6535	2341	66Ø5	2381
6346	2222	6416	2262	6466	23Ø2	6536	2342	66Ø6	2382
6347	2223	6417	2263	6467	23Ø3	6537	2343	66Ø7	2383
635Ø	2224	642Ø	2264	647Ø	23Ø4	654Ø	2344	661Ø	2384
6351	2225	6421	2265	6471	23Ø5	6541	2345	6611	2385
6352	2226	6422	2266	6472	23Ø6	6542	2346	6612	2386
6353	2227	6423	2267	6473	23Ø7	6543	2347	6613	2387
6354	2228	6424	2268	6474	23Ø8	6544	2348	6614	2388
6355	2229	6425	2269	6475	23Ø9	6545	2349	6615	2389
6356	223Ø	6426	227Ø	6476	231Ø	6546	235Ø	6616	239Ø
6357	2231	6427	2271	6477	2311	6547	2351	6617	2391
636Ø	2232	643Ø	2272	65ØØ	2312	655Ø	2352	662Ø	2392
6361	2233	6431	2273	65Ø1	2313	6551	2353	6621	2393
6362	2234	6432	2274	65Ø2	2314	6552	2354	6622	2394
6363	2235	6433	2275	65Ø3	2315	6553	2355	6623	2395
6364	2236	6434	2276	65Ø4	2316	6554	2356	6624	2396
6365	2237	6435	2277	65Ø5	2317	6555	2357	6625	2397
6366	2238	6436	2278	65Ø6	2318	6556	2358	6626	2398
6367	2239	6437	2279	65Ø7	2319	6557	2359	6627	2399

Machine Language To RIP Conversion Table

OCTAL	RIP	OCTAL	RIP	OCTAL	RIP	OCTAL	RIP	OCTAL	RIP
6630	2400	6700	2440	6750	2480	7020	2520	7070	2560
6631	2401	6701	2441	6751	2481	7021	2521	7071	2561
6632	2402	6702	2442	6752	2482	7022	2522	7072	2562
6633	2403	6703	2443	6753	2483	7023	2523	7073	2563
6634	2404	6704	2444	6754	2484	7024	2524	7074	2564
6635	2405	6705	2445	6755	2485	7025	2525	7075	2565
6636	2406	6706	2446	6756	2486	7026	2526	7076	2566
6637	2407	6707	2447	6757	2487	7027	2527	7077	2567
6640	2408	6710	2448	6760	2488	7030	2528	7100	2568
6641	2409	6711	2449	6761	2489	7031	2529	7101	2569
6642	2410	6712	2450	6762	2490	7032	2530	7102	2570
6643	2411	6713	2451	6763	2491	7033	2531	7103	2571
6644	2412	6714	2452	6764	2492	7034	2532	7104	2572
6645	2413	6715	2453	6765	2493	7035	2533	7105	2573
6646	2414	6716	2454	6766	2494	7036	2534	7106	2574
6647	2415	6717	2455	6767	2495	7037	2535	7107	2575
6650	2416	6720	2456	6770	2496	7040	2536	7110	2576
6651	2417	6721	2457	6771	2497	7041	2537	7111	2577
6652	2418	6722	2458	6772	2498	7042	2538	7112	2578
6653	2419	6723	2459	6773	2499	7043	2539	7113	2579
6654	2420	6724	2460	6774	2500	7044	2540	7114	2580
6655	2421	6725	2461	6775	2501	7045	2541	7115	2581
6656	2422	6726	2462	6776	2502	7046	2542	7116	2582
6657	2423	6727	2463	6777	2503	7047	2543	7117	2583
6660	2424	6730	2464	7000	2504	7050	2544	7120	2584
6661	2425	6731	2465	7001	2505	7051	2545	7121	2585
6662	2426	6732	2466	7002	2506	7052	2546	7122	2586
6663	2427	6733	2467	7003	2507	7053	2547	7123	2587
6664	2428	6734	2468	7004	2508	7054	2548	7124	2588
6665	2429	6735	2469	7005	2509	7055	2549	7125	2589
6666	2430	6736	2470	7006	2510	7056	2550	7126	2590
6667	2431	6737	2471	7007	2511	7057	2551	7127	2591
6670	2432	6740	2472	7010	2512	7060	2552	7130	2592
6671	2433	6741	2473	7011	2513	7061	2553	7131	2593
6672	2434	6742	2474	7012	2514	7062	2554	7132	2594
6673	2435	6743	2475	7013	2515	7063	2555	7133	2595
6674	2436	6744	2476	7014	2516	7064	2556	7134	2596
6675	2437	6745	2477	7015	2517	7065	2557	7135	2597
6676	2438	6746	2478	7016	2518	7066	2558	7136	2598
6677	2439	6747	2479	7017	2519	7067	2559	7137	2599

Machine Language To RIP Conversion Table

OCTAL	RIP	OCTAL	RIP	OCTAL	RIP	OCTAL	RIP	OCTAL	RIP
714Ø	26ØØ	721Ø	264Ø	726Ø	268Ø	733Ø	272Ø	74ØØ	276Ø
7141	26Ø1	7211	2641	7261	2681	7331	2721	74Ø1	2761
7142	26Ø2	7212	2642	7262	2682	7332	2722	74Ø2	2762
7143	26Ø3	7213	2643	7263	2683	7333	2723	74Ø3	2763
7144	26Ø4	7214	2644	7264	2684	7334	2724	74Ø4	2764
7145	26Ø5	7215	2645	7265	2685	7335	2725	74Ø5	2765
7146	26Ø6	7216	2646	7266	2686	7336	2726	74Ø6	2766
7147	26Ø7	7217	2647	7267	2687	7337	2727	74Ø7	2767
715Ø	26Ø8	722Ø	2648	727Ø	2688	734Ø	2728	741Ø	2768
7151	26Ø9	7221	2649	7271	2689	7341	2729	7411	2769
7152	261Ø	7222	265Ø	7272	269Ø	7342	273Ø	7412	277Ø
7153	2611	7223	2651	7273	2691	7343	2731	7413	2771
7154	2612	7224	2652	7274	2692	7344	2732	7414	2772
7155	2613	7225	2653	7275	2693	7345	2733	7415	2773
7156	2614	7226	2654	7276	2694	7346	2734	7416	2774
7157	2615	7227	2655	7277	2695	7347	2735	7417	2775
716Ø	2616	723Ø	2656	73ØØ	2696	735Ø	2736	742Ø	2776
7161	2617	7231	2657	73Ø1	2697	7351	2737	7421	2777
7162	2618	7232	2658	73Ø2	2698	7352	2738	7422	2778
7163	2619	7233	2659	73Ø3	2699	7353	2739	7423	2779
7164	262Ø	7234	266Ø	73Ø4	27ØØ	7354	274Ø	7424	278Ø
7165	2621	7235	2661	73Ø5	27Ø1	7355	2741	7425	2781
7166	2622	7236	2662	73Ø6	27Ø2	7356	2742	7426	2782
7167	2623	7237	2663	73Ø7	27Ø3	7357	2743	7427	2783
717Ø	2624	724Ø	2664	731Ø	27Ø4	736Ø	2744	743Ø	2784
7171	2625	7241	2665	7311	27Ø5	7361	2745	7431	2785
7172	2626	7242	2666	7312	27Ø6	7362	2746	7432	2786
7173	2627	7243	2667	7313	27Ø7	7363	2747	7433	2787
7174	2628	7244	2668	7314	27Ø8	7364	2748	7434	2788
7175	2629	7245	2669	7315	27Ø9	7365	2749	7435	2789
7176	263Ø	7246	267Ø	7316	271Ø	7366	275Ø	7436	279Ø
7177	2631	7247	2671	7317	2711	7367	2751	7437	2791
72ØØ	2632	725Ø	2672	732Ø	2712	737Ø	2752	744Ø	2792
72Ø1	2633	7251	2673	7321	2713	7371	2753	7441	2793
72Ø2	2634	7252	2674	7322	2714	7372	2754	7442	2794
72Ø3	2635	7253	2675	7323	2715	7373	2755	7443	2795
72Ø4	2636	7254	2676	7324	2716	7374	2756	7444	2796
72Ø5	2637	7255	2677	7325	2717	7375	2757	7445	2797
72Ø6	2638	7256	2678	7326	2718	7376	2758	7446	2798
72Ø7	2639	7257	2679	7327	2719	7377	2759	7447	2799

Machine Language To RIP Conversion Table

OCTAL	RIP	OCTAL	RIP	OCTAL	RIP	OCTAL	RIP	OCTAL	RIP
745Ø	28ØØ	752Ø	284Ø	757Ø	288Ø	764Ø	292Ø	771Ø	296Ø
7451	28Ø1	7521	2841	7571	2881	7641	2921	7711	2961
7452	28Ø2	7522	2842	7572	2882	7642	2922	7712	2962
7453	28Ø3	7523	2843	7573	2883	7643	2923	7713	2963
7454	28Ø4	7524	2844	7574	2884	7644	2924	7714	2964
7455	28Ø5	7525	2845	7575	2885	7645	2925	7715	2965
7456	28Ø6	7526	2846	7576	2886	7646	2926	7716	2966
7457	28Ø7	7527	2847	7577	2887	7647	2927	7717	2967
746Ø	28Ø8	753Ø	2848	76ØØ	2888	765Ø	2928	772Ø	2968
7461	28Ø9	7531	2849	76Ø1	2889	7651	2929	7721	2969
7462	281Ø	7532	285Ø	76Ø2	289Ø	7652	293Ø	7722	297Ø
7463	2811	7533	2851	76Ø3	2891	7653	2931	7723	2971
7464	2812	7534	2852	76Ø4	2892	7654	2932	7724	2972
7465	2813	7535	2853	76Ø5	2893	7655	2933	7725	2973
7466	2814	7536	2854	76Ø6	2894	7656	2934	7726	2974
7467	2815	7537	2855	76Ø7	2895	7657	2935	7727	2975
747Ø	2816	754Ø	2856	761Ø	2896	766Ø	2936	773Ø	2976
7471	2817	7541	2857	7611	2897	7661	2937	7731	2977
7472	2818	7542	2858	7612	2898	7662	2938	7732	2978
7473	2819	7543	2859	7613	2899	7663	2939	7733	2979
7474	282Ø	7544	286Ø	7614	29ØØ	7664	294Ø	7734	298Ø
7475	2821	7545	2861	7615	29Ø1	7665	2941	7735	2981
7476	2822	7546	2862	7616	29Ø2	7666	2942	7736	2982
7477	2823	7547	2863	7617	29Ø3	7667	2943	7737	2983
75ØØ	2824	755Ø	2864	762Ø	29Ø4	767Ø	2944	774Ø	2984
75Ø1	2825	7551	2865	7621	29Ø5	7671	2945	7741	2985
75Ø2	2826	7552	2866	7622	29Ø6	7672	2946	7742	2986
75Ø3	2827	7553	2867	7623	29Ø7	7673	2947	7743	2987
75Ø4	2828	7554	2868	7624	29Ø8	7674	2948	7744	2988
75Ø5	2829	7555	2869	7625	29Ø9	7675	2949	7745	2989
75Ø6	283Ø	7556	287Ø	7626	291Ø	7676	295Ø	7746	299Ø
75Ø7	2831	7557	2871	7627	2911	7677	2951	7747	2991
751Ø	2832	756Ø	2872	763Ø	2912	77ØØ	2952	775Ø	2992
7511	2833	7561	2873	7631	2913	77Ø1	2953	7751	2993
7512	2834	7562	2874	7632	2914	77Ø2	2954	7752	2994
7513	2835	7563	2875	7633	2915	77Ø3	2955	7753	2995
7514	2836	7564	2876	7634	2916	77Ø4	2956	7754	2996
7515	2837	7565	2877	7635	2917	77Ø5	2957	7755	2997
7516	2838	7566	2878	7636	2918	77Ø6	2958	7756	2998
7517	2839	7567	2879	7637	2919	77Ø7	2959	7757	2999