

COMPANY CONFIDENTIAL

CONUS 9100

DIGITAL COMPUTER SERIES

MANAGEMENT REPORT

FOR PLANNING PURPOSES ONLY

SUBJECT TO CHANGE WITHOUT
NOTICE

Prepared By:

Government Marketing Planning
St. Paul, Minnesota

Date:

September 14, 1964

COMPANY CONFIDENTIAL

Processor concepts set forth in this report are the joint effort of the following:

D. H. Anderson

J. P. Fourre

R. T. McArdell

P. D. Byrns

SYNOPSIS

This report describes the characteristics of the CONUS 24 Computer line which has also been referred to as the 9100 Digital Computer series. A total of eight processors are described in terms of their basic features and instruction repertoires. These computers are upward program compatible. Also described are a number of optional arithmetic and control units which can be used to tailor the processors to specific applications. It is essential that a plan such as this be accepted as a part of our CONUS 24 announcement to avoid the development of another generation of non-compatible computers. If this plan is acceptable, UNIVAC will automatically begin work on an integrated line without delay.

TABLE OF CONTENTS

<u>SECTION</u>	<u>CONTENTS</u>	<u>PAGE</u>
A	Management Introduction To the CONUS 24 Computer Series	1
B	CONUS 24 Processor Specifica- tions	4
C	Programmers Reference	10
D	CONUS Escape Mode	27
APPENDIX A	Summary of Specifications	
APPENDIX B	Logical Element Instruction Subset	

A. MANAGEMENT INTRODUCTION TO THE CONUS 24 COMPUTER SERIES

1. The CONUS 24 Computer Series is a compatible line of eight internally programmed data processors. These are five 24 bit word and three 48 bit word processors as follows:

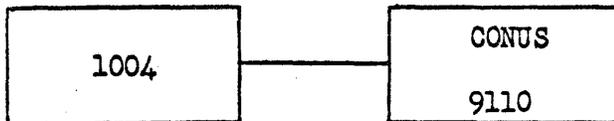
9110	(24 bit word)
9120	(24 bit word)
9130	(24 bit word)
9140	(24 bit word)
9150	(24 bit word)
9160	(48 bit word)
9170	(48 bit word)
9180	(48 bit word)

2. The design objectives of the CONUS 24 series are of prime importance to management and are listed as follows:
 - a. To protect the 1,500 plus, 1004 installations, by providing an internally programmed unit for direct connection to 1004 with no hardware 1004 modifications. This is the CONUS 9110 Processor which is basically 3 to 4 times as fast as the IBM Model 30, which could rent for approximately 1/2 to 2/3 of the Model 30. This unit will allow the 1004 user to run all his current programs and at the same time expand into an internally programmed processor which has the computer capability to grow beyond the size of an average 490 installation.
 - b. To provide an internally programmed processor to be sold where a 1004 cannot because of its externally programmed features. This processor, complete with basic card and printer peripherals, to be reproduced at a manufacturing cost which would allow rental at 1004 prices. This basic processor is the CONUS 9110 which is upward compatible with the 7 higher performance CONUS processors.
 - c. To provide a series of processors which can easily accept current data codes such as XS-3, Fieldata, and IBM/BCD which will have predominance for the next 2 to 3 years and at the same time, with no restrictions be 100% compatible with the new ASC II code which must be a minimum requirement for any new computer line.
 - d. To provide a series of processors which are an excellent real-time communications processor, an excellent batch tape data processor, an excellent scientific data processor, with any of the above on any combination on any processor.
 - e. To provide a series of processors for which the software development is simplified from existing UNIVAC Computers. These items are covered in detail under Section D (Special Features).

- f. To provide a series of processors which are superior to competition from both price and performance. These processors have no built in restrictions and may be improved with advances in circuit and memory science and as the competition dictates.
 - e. To eliminate the gross inefficiency of manufacturing, selling, supporting, software development and maintaining complete independent and unrelated systems such as the 1050, UNIVAC III, 418, 490, and 1108.
3. Of prime importance to the CONUS 24 series is the concept under which arithmetic units will be provided. Any of the CONUS 24 processors mentioned previously, would be thought of as a logical element with an extreme high degree of communications and data processing capability, but a minimum scientific (not restricted) ability. Except for the 9110 Model, each processor has optionally, a range of 8 different arithmetic and executive control units which may be interfaced to any given processor. These optional units are as follows:
- a. 24 bit fixed point, low cost, medium speed arithmetic unit. Includes multiply/divide and supporting logical commands.
 - b. 24 bit fixed point, higher cost, high speed arithmetic unit. Includes multiply/divide and supporting logical commands (two types).
 - c. 48 bit fixed point/floating point, medium price/medium performance, includes double precision add, subtract, multiply and divide. The floating point includes a variable base to provide a greater range with better accuracy than the IBM System/360.
 - d. 48 bit fixed point/floating point, higher cost high performance, includes double precision add, subtract, multiply, and divide. The floating point includes a variable base to provide a greater range with better accuracy than the IBM System/360 (two types).
 - e. Decimal/Edit Set. This optional arithmetic unit provides the decimal and editing commands for those users who require a high performance data processing installation.
 - f. Executive Control Set. This is an optional control set which provides executive (hardware) control over Input/Output devices, floating operand registers, memory lockout and interrupt masks. This set is provided for users who require absolute protection of critical programs, high speed dynamic loading and high speed Input/Output.
4. In this report, under Section D, "Escape Class and Special Features", detail definition of the above optional arithmetic and executive features are provided. This concept allows the UNIVAC Marketing

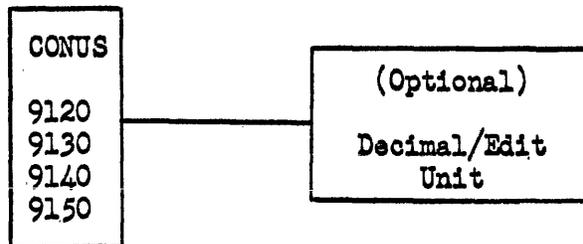
organization to custom fit a computer system to any given application. Following are typical computer systems configurations for major classes of data processing.

a. Existing 1004 installations:



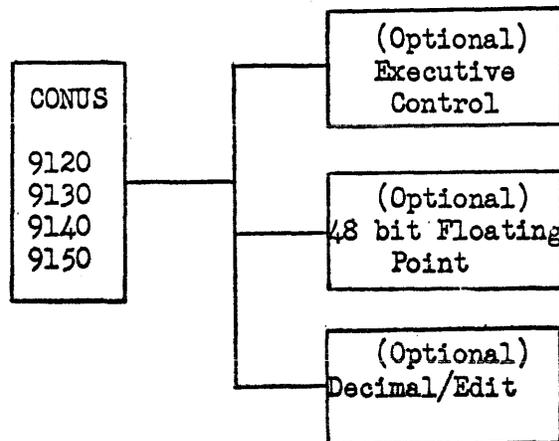
- a.1 CONUS 9110 connects directly to 1004 without modification, existing programs run without alteration.
- a.2 CONUS 9110 provides at least 10 times additional computer capability over the 1004.

b. Real-time Communications and High Volume Batch Data Processing Installation:



- b.1 The optional Decimal/Edit Unit is provided to assist batch data processing. It is not required for the communications real-time areas.

c. Critical Real-Time Communications, High Volume Batch and Scientific Data Processing Installation:



B. THE CONUS 24 PROCESSOR SPECIFICATIONS

As mentioned under Section A, the CONUS 24 series consists of eight processors, five 24 bit word machines and three 48 bit word machines. Listed below are the eight processors in ascending sequence, by capability:

9110
9120
9130
9140
9150
9160
9170
9180

1. CONUS 9110

The CONUS 9110 is specifically designed to complement the 1004 and to provide a competitive base system from which to sell an integrated product line.

9110 Specifications

- a. Memory cycle time = 2.4 microseconds.
- b. Word size or address structure = 24 bit word.
- c. Basic addressable units of data = 8 to 24 bits.
- d. Minimum memory size = 4,096 words
= 12,288 - 8 bit characters.
- e. Memory increments = 4,096 words
= 12,288 - 8 bit characters.
- f. Maximum memory = 32,768 words
= 98,304 - 8 bit characters.
- g. Index registers = 3
- h. Indirect addressing - YES
- i. Amount of memory directly addressable = 32,768 words.
- j. Average instruction time = 7.2→9.6 microseconds.
- k. Number of instructions or Functions Codes = 35.
- l. Minimum number of input/output channels = 1.
- m. Maximum number of input/output channels = 3.
- n. ESCAPE MODE with Multiple Arithmetic units - NO.

NOTE:

Function Codes are listed under Appendix B. The 9110 includes Function Codes 00 - 42 octal, excluding Functions 24→27 and 43. 52→56 are provided through escape mode.

2. CONUS 9120

The CONUS 9120 is the second upward compatible processor.

9120 Specifications

- a. Memory cycle time = 2.4 microseconds. .
- b. Word size or address structure = 24 bit word.
- c. Basic addressable units of data = 8 to 24 bits.
- d. Minimum memory size = 4,096 words
= 12,288 - 8 bit characters.
- e. Memory increments = 4,096 words
= 12,288 - 8 bit characters.
- f. Maximum memory = 65,536 words
= 196,608 - 8 bit characters.
- g. Index registers = 3.
- h. Indirect addressing - YES
- i. Amount of memory directly addressable = 32,768 words.
- j. Average instruction time 7.2→9.6 microseconds.
- k. Number of instructions or Functions Codes = 41 + optional arithmetic and control.
- l. Minimum number of input/output channels = 3.
- m. Maximum number of input/output channels = 6.
- n. Escape mode with multiple arithmetic units - YES.

NOTE:

The 9120 may have an option of 7 different arithmetic units plus an optional executive control unit. Please reference Section D of this report for detailed description of Escape Mode and optional Function Code list.

3. CONUS 9130

The CONUS 9130 is the third upward compatible processor. The 9130 is approximately 3 times as fast as the 9120.

9130 Specifications

- a. Memory cycle time = 2 microseconds.
- b. Word size or address structure = 24 bit word
- c. Basic addressable units of data = 8 and 24 bits.
- d. Minimum memory size = 8,192 words
= 24,576 - 8 bit characters.
- e. Memory increments = 8,192 words.
- f. Maximum memory = 65,536 words
= 196,608 - 8 bit characters.
- g. Index registers = 3.
- h. Indirect addressing - YES.
- i. Amount of memory directly addressable = 32,768 words.
- j. Average instruction time = 3 microseconds.
- k. Number of instructions or Function Codes = 41 plus optional arithmetic and control.
- l. Minimum number of input/output channels = 4.
- m. Maximum number of input/output channels = 16.
- n. Escape Mode with multiple arithmetic units - YES.

NOTE:

The 9130 may have as an option, 7 different arithmetic units, plus an optional executive control unit. Please reference Section D of this report for detail description of Escape Mode and optional Function Code list.

4. CONUS 9140

The CONUS 9140 is the fourth upward compatible processor. The 9140 is twice as fast as the 9130.

9140 Specifications

- a. Memory cycle time = .75 microseconds

- b. Word size or address structure = 24 bit word.
- c. Basic addressable units of data = 8 and 24 bits.
- d. Minimum memory size = 8,192 words
= 24,576 - 8 bit characters.
- e. Memory increments = 8,192 words.
- f. Maximum memory = 65,536 words
= 196,608 - 8 bit characters.
- g. Index register = 3.
- h. Indirect addressing - YES.
- i. Amount of memory directly addressable = 32,768 words.
- j. Average instruction time = 1.2 microseconds.
- k. Number of instructions or Function Codes = 41 plus optional arithmetic and control.
- l. Minimum number of input/output channels = 4
- m. Maximum number of input/output channels = 16
- n. Escape Mode with multiple arithmetic units - YES.

NOTE:

The 9140 may have as an option, 7 different arithmetic units, plus an optional executive control unit. Please reference Section D of this report for detail description of Escape Mode and optional Function Code list.

5. CONUS 9150

The CONUS 9150 is the fifth upward compatible processor. The 9150 is twice as fast as the 9140.

9150 Specifications

- a. Memory cycle overlapped/effective = .375 microseconds.
- b. Word size or address structure = 24 bit word.
- c. Basic addressable units of data = 8 and 24 bits.
- d. Minimum memory size = 8,192 words
= 24,576 - 8 bit characters.

- e. Memory increments = 8,192 words.
- f. Maximum memory = 131,072.
- g. Index registers = 3.
- h. Indirect addressing - YES.
- i. Amount of memory directly addressable = 32,768 words.
- j. Average instruction time = .75 microseconds.
- k. Number of instructions or Function Codes = 41 plus optional arithmetic and control.
- l. Minimum number of input/output channels = 4.
- m. Maximum number of input/output channels = 16.
- n. Escape Mode with multiple arithmetic units - YES.

NOTE:

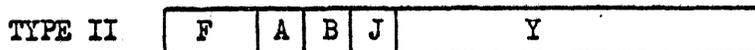
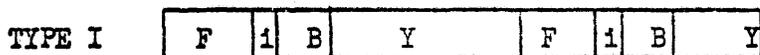
The 9150 may have as an option, 7 different arithmetic units, plus an optional executive control unit. Please reference Section D of this report for detailed description of Escape Mode and optional Function Code list.

6. CONUS 9160

The CONUS 9160 is the sixth upward compatible processor. It is the first 48 bit processor in the CONUS series.

9160 Specifications

The 9160 Instruction Repertoire includes the Master Instruction Set which is provided for the 24 bit CONUS processors. Programs written for the 9110, 9120, 9130, 9140, and 9150 will run on the 9160. Instruction format for the 9160 is as follows:



The Type I Instruction Format consists of two 24 bit instructions per 48 bit word. It is the same as the normal 24 bit format.

The Type II format is as follows:

F = 12 bit Function Code ($2^{36} - 2^{47}$) biased to octal value 7700.

A = 4 bit general register designator ($2^{32} - 2^{35}$) specifies one of 16 general purpose accumulators.

B = 4 bit index register designator, ($2^{28} - 2^{31}$) specifies one of 16 index registers.

J = 4 bit operand interpretation designator ($2^{24} - 2^{27}$) currently not assigned.

Y = Operand address ($2^0 - 2^{23}$), may either be an operand or operand address.

a. Memory cycle time overlapped/effective = .375 microseconds.

b. Minimum memory = 32,768 words.
= 196,608 - 8 bit characters.

c. Memory increments = 32,768 words.

d. Maximum memory = 262,144 words.

e. Minimum I/O channels = 4.

f. Maximum I/O channels = 24.

g. Average 24 bit instruction time = .75 microseconds.

h. Average 48 bit instruction time = .75 microseconds.

NOTE:

Individual 48 bit function codes have not been assigned at this time.

7. CONUS 9170

The CONUS 9170 is the seventh upward compatible processor.

CONUS 9170 Specifications

The CONUS 9170 is functionally the same as the 9160. The 9170 is dependent on the development of .4 microsecond memory and the necessary logic to work at these speeds.

8. CONUS 9180

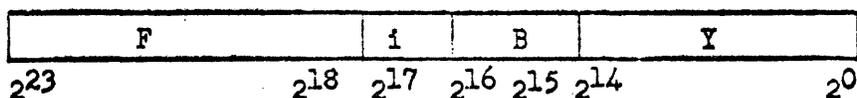
The CONUS 9180 is currently beyond the capability of present UNIVAC know-how and still remain compatible. An answer to this processor is the NIKE X, which is a special military processor, but could have many commercial applications.

C. CONUS 24 BIT WORD INSTRUCTION FORMAT AND PROGRAMMER'S REFERENCE

This section includes the Instruction Word Format for the 24 bit word processors, followed by a narrative description of the Instruction Rapertoire. Function Codes 53, 54, 55, and 56 are in the Escape Mode (Executive Control Set), which requires the processor to be initialized for input/output commands.

1. Instruction Format

The instruction format of the CONUS 24 is as follows:



Where:

F = 6 bit function code designator, specifies the individual processor instruction.

i = 1 bit indirect address designator;

B = 2 bit index register designator;

Y = 15 bit memory address designator, contains initial operand address. In certain instructions appears as a mask or constant.

2. Control and Operational Registers

In the CONUS 24 series, the following control and operational registers may be addressed by two methods.

R = General Register
P = Program Address Counter
B¹ = B Register 1
B² = B Register 2
B³ = B Register 3

- a. The first method of addressing is implied within the function codes.
- b. In the first two processors of the CONUS 24 series, the R, P, and B registers are in memory and become Y addressable. These registers occupy the following memory addresses:

P = 00000
R = 00001
B₁ = 00005
B₂ = 00006
B₃ = 00007

3. Indirect Address Designator

The indirect address designator applies to all instructions except those individually specified in the instruction repertoire. B register designators or indexing is allowed for all instructions except those which imply B registers in the function codes. When indirect addressing is used, the processor will continue to cascade indirect addresses until the indirect bit is not set. Indexing is performed on every indirect address unless the B field is zero.

4. Instruction Repertoire - Logical Element

Following is the instruction repertoire for the CONUS 24 Logical Element. Function code narratives are provided for function values 00 - 52. Function codes 53 through 77 are included in the Escape sets. The following conventions apply.

- a. All times are shown in microseconds, which includes the first two CONUS 24 Computers, 9110 and 9120. For every indirect and B register reference for the 9110 and 9120 an additional 2.4 microseconds must be added to times shown.
- b. y = the operand address field of the instruction as stored in memory.
- c. Y = the contents of memory location y .
- d. $\bar{Y} = y + B_p$, specifies the contents of memory location $y + B_p$.
- e. iY specifies the contents of the indirect address. y is the address of the address.
- f. $i\bar{Y}$ specifies the contents of the indirect address. \bar{Y} is the address of the address.

Function Code: 00

Operation: Stop, processor comes to a halt, initiated input/output transfers are terminated.

Designators: Hold

Time: 4.8

Function Code: 01

Operation: Store Program Status Code. The contents of the machine designators are stored in memory as specified by the following forms:

\bar{Y}
 iY
 $i\bar{Y}$

Designators: Clear

Time: 7.2

Function Code: 02

Operation: Load Program Status Codes, the processor designators are loaded by one of the following operand types.

Y The low order 15 bits (y) of the instruction are used to specify a memory location whose contents fill the processor designators according to master bit format.

\bar{Y} $B_p + y$ are added to form the address whose contents fill the processor designators according to master bit format.

iY Y is used to indirectly address the contents of a memory location which fill the processor designators according to master bit format.

$i\bar{Y}$ \bar{Y} is used to indirectly address the contents of a memory location which fill the processor designators according to master bit format.

Master bit format:

2^0 = Positive	2^4 = Non-Zero (ignored)
2^1 = Zero	2^5 = No-Carry (ignored)
2^2 = Carry	2^6 = Overflow
2^3 = Negative (ignored)	

Note:

Respective bit positions must be one to set Positive, Zero, Carry, and Overflow designators. Negative, Non-Zero and No-Carry are shown here to follow format of master skip instruction, there are no processor designators for these conditions and cannot be affected by this instruction.

Designators: Fill

Time: 7.2

Function Code: 03

Operation: Execute Remote Y, the instruction stored at location Y is executed. The following forms of Y apply:

Y
 \bar{Y}
iY
 $i\bar{Y}$

Designators: Dependent on Remote Instruction executed.

Time: 7.2 plus execution time of Remote Instruction.

Function Code: 04

Operation: Execute Remote Y - Set Return, the contents of $P + 1$ are stored in memory location Y, only bit positions $2^0 - 2^{14}$ of Y are changed, Y must always be even. $Y + 1$ is executed which must be an unconditional Jump to $Y + 2$ to remain in the remote sequence. The following forms of Y apply:

\bar{Y}
 \bar{Y}
 iY
 $i\bar{Y}$

Designators: Dependent on Remote Instruction executed.

Time: 7.2 plus execution time of Remote Instruction.

Function Code: 05

Operation: Skip on Y Mask, the next sequential instruction is skipped if the condition or conditions being tested by Y are met. The following forms of Y apply:

- y y the low order 15 bits are the direct mask for testing skip conditions.
- \bar{Y} The contents of $B_0 + y$ are added to form \bar{Y} which is then the direct mask for testing skip conditions.
- iY Y is used as an address to address the contents of a memory location which contains the mask for testing skip conditions.
- $i\bar{Y}$ The contents of Y are added to B_0 to form \bar{Y} which then is used as an address to address the contents of a memory location which contains the mask for testing skip conditions.

Note:

To test for a processor designator, the mask must have the respective bit position set to one as follows:

- | | |
|------------------|---------------------|
| 2^0 = Positive | 2^4 = Non-Zero |
| 2^1 = Zero | 2^5 = No-Carry |
| 2^2 = Carry | 2^6 = No-Overflow |
| 2^3 = Negative | |

One or any number of conditions may be tested, the Skip Instruction is an OR function.

Designators: Hold

Time: 4.8 - No Skip
7.2 - Skip

Function Code: 06

Operation: Jump to Y, an unconditional Jump is made to address Y. The following forms of Y apply.

Y
 \bar{Y}
iY
i \bar{Y}

Designators: Hold

Time: 4.8

Function Code: 07

Operation: Set/Clear Lockout, Interrupt lockout is set if $Y 2^0 = 1$, Interrupt lockout is cleared if $Y 2^0 = 0$. The following forms of Y apply:

y The 15 bits of y are the direct mask for setting or clearing interrupt lockout.

\bar{Y} The contents of $B_0 + y$ are added to form \bar{Y} which is used as the direct mask for setting or clearing interrupt lockout.

iY The 15 bits of Y are used as an address to address a memory location which contains the mask for setting or clearing interrupt lockout.

Note:
After the execution of clear lockout, interrupts are held for one additional instruction.

Designators: Hold

Time: 4.8

Function Code: 10

Operation: Add $(R) + (Y) \rightarrow R$, the contents of memory location Y are added to the general register R. The following forms of Y apply:

Y
Y
iY
iY

Designators: Fill

Time: 9.6

Function Code: 11

Operation: Subtract $(R) - (Y) \rightarrow R$, the contents of memory location Y are subtracted from the general register R, the difference is left in R. The following forms of Y apply:

Y
Y
iY
iY

Designators: Fill

Time: 9.6

Function Code: 12

Operation: And $(R) \odot (Y) \rightarrow R$, the Logical Product of the contents of memory location Y and the contents of general register R are formed with the result left in R. The following forms of Y apply.

Y
Y
iY
iY

Designators: Fill

Time: 9.6

Function Code: 13

Operation: Exclusive OR $(R) \oplus (Y) \rightarrow R$, the selective complement of the mask in general register R is used to selectively complement the contents of operand Y. The result is in R, the contents of Y are unchanged. The following forms of Y apply:

Y
 \bar{Y}
iY
i \bar{Y}

Designators: Fill

Time: 9.6

Function Code: 14

Operation: Store $(B)_b$ at Y, the contents of the specified B register are stored at the contents of memory location Y. Y is the only form of memory addressing that may be used.

Designators: Hold

Time: 9.6

Function Code: 15

Operation: Load B_b from Y. The contents of memory location Y are loaded into the specified B register. Y is the only form of memory addressing that may be used.

Designators: Hold

Time: 9.6

Function Code: 16

Operation: $B + \text{constant}$, $(B)_b + y \rightarrow B_b$, the low order 15 bits of the instruction which are y are added to the contents of the specified B register, the result of the addition is stored into B. There is no direct or indirect operand reference for this instruction.

Designators: Hold

Time: 7.2

Function Code: 17

Operation: Compare $B (B)_b - (Y)$, skip next instruction if zero designator set, $B_b + 1 \rightarrow B$, the contents of the memory location specified by Y are compared to the contents of the specified B register, setting the appropriate designator. The next instruction is skipped if zero designator is set. If not, plus 1 is added to B_b and the result is stored in B_b . y is the only form of operand allowed.

Designators: Fill

Time: 9.6 - No Skip
7.2 - Skip

Function Code: 20

Operation: Load R with Y, the contents of memory location Y are loaded into the general Register R. The following forms of Y apply:

\overline{Y}
 \overline{Y}
 $i\overline{Y}$
 $i\overline{Y}$

Designators: Fill

Time: 9.6

Function Code: 21

Operation: Load R with $(Y)_U$, the upper eight bits ($2^{16} - 2^{23}$) of the contents of memory location Y are loaded into the lower eight bits ($2^0 - 2^7$) of the general register R. $2^8 - 2^{23}$ of R are cleared to 0's. The following forms of Y apply:

\overline{Y}
 \overline{Y}
 $i\overline{Y}$
 $i\overline{Y}$

Designators: Fill

Time: 9.6

Function Code: 22

Operation: Load R with $(Y)_M$, the middle eight bits ($2^8 - 2^{15}$) of the contents of memory location Y are loaded into the lower eight bits ($2^0 - 2^7$) of general register R. $2^8 - 2^{23}$ of R are cleared to 0's. The following forms of Y apply:

\overline{Y}
 \overline{Y}
 $i\overline{Y}$
 $i\overline{Y}$

Designators: Fill

Time: 9.6

Function Code: 23

Operation: Load R with $(Z)_L$, the low order eight bits ($2^0 - 2^7$) of the contents of memory location Y are loaded into the low order eight bits ($2^0 - 2^7$) of the general register R. $2^8 - 2^{23}$ of R are cleared to 0's.

Designators: Fill

Time: 9.6

Function Code: 24

Operation: Compare $(R) - (Y)$, the contents of the general register R are compared with the contents of Y, the appropriate processor designators are set. Neither the contents of R or Y are changed.

$\frac{Y}{\bar{Y}}$
 $\frac{iY}{i\bar{Y}}$

Designators: Fill

Time: 9.6

Function Code: 25

Operation: Compare $(R) - (Y)_U$, the contents of general register R are compared to the high order eight bits ($2^{16} - 2^{23}$) of memory location Y. The appropriate processor designators are set. Neither the contents of R or Y_U are changed. The following forms of Y apply:

$\frac{Y}{\bar{Y}}$
 $\frac{iY}{i\bar{Y}}$

Designators: Fill

Time: 9.6

Function Code: 26

Operation: Compare $(R) - (Y)_M$, the contents of general register R are compared to the high order eight bits ($2^8 - 2^{15}$) of memory location Y. The appropriate processor designators are set. Neither the contents of R or Y_M are changed. The following forms of Y apply:

Y
Y
iY
iY

Designators: Fill

Time: 9.6

Function Code: 27

Operation: Compare (R) - (Y)_L, the contents of the general register R are compared to the low order eight bits of memory (2⁰ - 2⁷). The appropriate processor designators are set. Neither the contents of R or Y are changed. The following forms of Y apply:

Y
Y
iY
iY

Designators: Fill

Time: 9.6

Function Code: 30

Operation: Store (R) at Y, the contents of the general register R are stored at memory location Y. The following forms of Y apply:

Y
Y
iY
iY

Designators: Hold

Time: 9.6

Function Code: 31

Operation: Store (R)_L at Y_H, the low order eight bits (2⁰ - 2⁷) of the general register R are stored in the eight high order bits (2¹⁶ - 2²³) of Y. Bit positions 2⁰ - 2¹⁵ of Y are not changed. The following forms of Y apply:

Y
Y
iY
iY

Designators: Hold

Time: 9.6

Function Code: 32

Operation: Store $(R)_L$ at Y_M , the low order eight bits ($2^0 - 2^7$) of the general register R are stored in the middle eight bits ($2^8 - 2^{15}$) of memory location Y. Bit positions ($2^0 - 2^7$) and ($2^{16} - 2^{23}$) are not changed. The following forms of Y apply:

Y
 \bar{Y}
iY
i \bar{Y}

Designators: Hold

Time: 9.6

Function Code: 33

Operation: Store $(R)_L$ at Y_L , the low order eight bits ($2^0 - 2^7$) of the general register R are stored in the low eight bits ($2^0 - 2^7$) of memory location Y. Bit positions $2^8 - 2^{23}$ are not changed. The following forms of Y apply;

Y
 \bar{Y}
iY
i \bar{Y}

Designators: Hold

Time: 9.6

Function Code: 34

Operation: Increment (Y), the contents of memory location Y are incremented by 1. The following forms of Y apply:

Y
 \bar{Y}
iY
i \bar{Y}

Designators: Fill

Time: 7.2

Function Code: 35

Operation: Enter R with Constant, the fifteen bits of y $2^0 - 2^{14}$ are loaded into the general register R. Bits $2^{15} - 2^{23}$ of R are sign filled. The following are forms of Y.

y The fifteen bits of the instruction which is the constant.

\bar{Y} $y + B_p$ are added to form \bar{Y} which is the constant.

iY Y is the address at which is store the 15 least significant ($2^0 - 2^{14}$) bits which form the constant.

$i\bar{Y}$ \bar{Y} is the address at which is stored the 15 least significant bits which form the constant.

Designators: Fill

Time: 7.2

Function Code: 36

Operation: Decrement (Y), skip if negative, the contents of Y are decremented by 1. If Y final is negative, skip the next instruction. If Y final is not negative, execute the next sequential instruction. The following forms of Y apply:

Y
 \bar{Y}
 iY
 $i\bar{Y}$

Designators: Hold

Time: 7.2 - No Skip
9.6 - Skip

Function Code: 37

Operation: Store Address, the low order 15 bits ($2^0 - 2^{14}$) of the general register R are stored at memory address Y. Bit positions $2^{15} - 2^{23}$ of Y are not changed. The following forms of Y apply:

Y
 \bar{Y}
 iY
 $i\bar{Y}$

Designators: Hold

Time: 9.6

Function Code: 40

Operation: Shift Storage left circular 8, the contents of memory location Y are left circular shifted 8 binary places. The following forms of Y apply:

Y
 \bar{Y}
iY
i \bar{Y}

Designators: Fill

Time: 7.2

Function Code: 41

Operation: Shift R right K, the general register R is shifted right open K places. K may have a value of 0 - 7. The high order bits of R sign fill. K may take the following form:

y The least significant 3 bits of y form the value K.

\bar{Y} The value of B_p is added to y to form \bar{Y} whose least significant 3 bits form the value K.

iY The value of Y is an address whose contents (3 least significant bits) form the value K.

Designators: Fill

Time: 4.8 + 2.4 (K)

Function Code: 42

Operation: Shift R left K, the contents of the general register R are left shifted open by the value K. Least significant bit positions of R are zero filled. K may have a value of 0 - 7 and take the following form:

y The least significant 3 bits of y form the value K.

\bar{Y} The value of B_p is added to y to form \bar{Y} whose 3 least significant bits) form the value K.

iY The value of Y is an address whose contents (3 least significant bits) form the value K.

i \bar{Y} The value of \bar{Y} is an address whose contents (3 least significant bits) form the value K.

Designators: Fill

Time: 4.8 + 2.4 (K)

Function Code: 43

Operation: Parity on (Y), Set Carry Des. if even, set plus and zero, the contents of memory location Y are tested for parity, if even carry designator is set. Set the appropriate plus, zero designator for test on Y. The following forms of Y apply:

$$\begin{array}{c} Y \\ \bar{Y} \\ iY \\ i\bar{Y} \end{array}$$

Designators: Fill

Time: 7.2

Function Code: 44

Operation: Reserved function code, execution will cause interrupt to location 111.

Function Code: 45

Operation: Reserved function code, execution will cause interrupt to location 113.

Function Code: 46

Operation: Reserved function code, execution will cause interrupt to location 115.

Function Code: 47

Operation: Reserved function code, execution will cause interrupt to location 117.

Function Code: 50

Operation: Reserved function code, execution will cause interrupt to address 121.

Function Code: 51

Operation: Reserved function code, execution will cause interrupt to location 123.

Function Code: 52

Operation: ESCAPE to Y, Function Codes 61 through 77 are reordered according to mode specified by Y. This command dynamically connects the required arithmetic or Special Control Unit.

The Y mask format is as follows:

- 2^0 = 24 bit fixed point arithmetic
- 2^1 = 48 bit fixed point/floating point
- 2^2 = Decimal/Edit
- 2^3 = High Speed I/O Control
- 2^4 = Executive Control Set
- 2^5 = Reserved

Y Mask may take the following form:

- y The 15 bits of y ($2^0 - 2^{14}$) are the mask.
- \bar{Y} The contents of B_0 are added to y to form \bar{Y} which is the mask.
- iY The value of Y is an address whose contents ($2^0 - 2^{14}$) are the mask.
- i \bar{Y} The value of \bar{Y} is an address whose contents ($2^0 - 2^{14}$) are the mask.

Designators: Hold

Time: 4.8

Function Code: 53

Operation: Activate Input Channel, the input channel as specified by the Y mask is set active. Channel mask format is:

- 2^0 = Channel 0
- 2^1 = Channel 1
- 2^2 = Channel 2

The corresponding bit position must be set to 1 to activate the respective input channel. Up to 15 input channels may be identified with bits $2^0 - 2^{14}$ of Y mask. Y mask has the following formats:

- y The value of y itself is the mask.
- \bar{Y} The value of B_0 is added to y to form \bar{Y} which is the mask.
- iY The value of Y is an address whose contents ($2^0 - 2^{14}$) for the mask.

$i\bar{Y}$ The value \bar{Y} is an address whose contents ($2^0 - 2^{14}$) from the mask.

Function Code: 54

Operation: Activate output channel, the output channel as specified by Y mask is set active. Channel mask format is:

2^0 = Channel 0
 2^1 = Channel 1
 2^2 = Channel 2

The corresponding bit position must be to activate the respective output channel. Up to 15 output channels may be identified with bits $2^0 - 2^{14}$ of Y mask. Y mask has the following formats.

y The value of y itself is the mask.

\bar{Y} The value of B_D is added to y to form \bar{Y} which is the mask.

iY The value of Y is an address whose contents ($2^0 - 2^{14}$) form the mask.

$i\bar{Y}$ The value \bar{Y} is an address whose contents ($2^0 - 2^{14}$) form the mask.

Designators: Hold

Time: 4.8

Function Code: 55

Operation: External function (R) Y_N , the contents of the general Register R are forced out channel Y_N as an external function. Y is a translated value which specifies the corresponding output channel. Y may have the following forms:

y The value y is translated for channel number.

\bar{Y} The value of B_D is added to y to form \bar{Y} which is translated for channel number.

iY The value Y is an address whose contents ($2^0 - 2^{14}$) are translated for channel number.

$i\bar{Y}$ The value \bar{Y} is an address whose contents ($2^0 - 2^{14}$) are translated for channel number.

Designators: Hold

Time: 7.2

Function Codes: 53 - 77

Operation: Described in the Escape Mode sets for optional arithmetic units.

D. THE CONUS ESCAPE MODE

This section deals with the optional arithmetic and control units which are identified through the Escape Instruction. Function Code assignments are provided for each Escape Set, with machine timing provided for the fixed and floating point multiply and divide. Individual function code timing will be provided upon request. Following are the optional Escape Sets.

1. Executive Control
2. High Speed Input/Output
3. Decimal/Edit
4. Fixed Point 24 Bit, 3 groups
5. Fixed/Floating Point 48 bit, 3 groups.

1. Executive Control Escape Set

Function Codes 53 through 56 are standard on the 9110, 9120, 9130, 9140, and 9150. Function Codes 53 through 64 are standard on the 9130, 9140, and 9150.

Programs may use Function Codes in this set by first executing the Escape Instruction (52) with the proper mask.

Function Code

53	Activate Input Channel by Y mask	Hold
54	Activate Output Channel by Y	Hold
55	EF, (R) \rightarrow Y_N (translate Y_N)	Hold
56	Store ESI, Chan. Y_N , Ack. Y_N (translate Y)	Hold
57	Load F_1 with (Y)	Hold
60	Load F_2 with (Y)	Hold
61	Store F_1 at Y	Hold
62	Store F_2 at Y	Hold
63	Load MLR with (Y)	Hold
64	Load Interrupt Mask with (Y)	Hold

- a. F_1 = Float Register 1, this register is used to provide dynamic relocation of programs and subroutines. The contents of F_1 are added to Y for operand reference allowing program execution from any memory location without prior modification of instructions.
- b. F_2 = Float Register 2, this register is provided with F_1 in a two bank processor to allow dynamic relocation of programs independent of data.
- c. MLR = Memory Lockout Register, this is used to provide protection of executives and critical real-time programs.
- d. Interrupt Mask = Interrupt mask register is used to temporarily lock out non-critical interrupts during periods of critical interrupt processing.

2. High Speed Input/Output Control

The High Speed Input/Output Control Set provides either 2 cycle or 1 cycle Input/Output transfers. High Speed Input/Output transfers are limited to 4,096 word blocks. There are optionally 1 to 4 high speed I/O channels available. One cycle Input/Output is provided by incorporating hardware Buffer Control Addresses and Buffer Counts. Two cycle Input/Output is provided by incorporating hardware Buffer Control Address register only. These registers are controlled by the following function codes in this Escape Class.

<u>Function Code</u>	<u>Operation</u>
53	Load Input Channel Zero Control Register, (registers) with the contents of Y. If Y is even, the Buffer Base Address Register is loaded, if Y is odd, the buffer count register is loaded.
54	Load Output Channel Zero Control Register (registers) with the contents of Y. If Y is even, the buffer base address register is loaded, if y is odd, the buffer count register is loaded.
55	Same as above, Channel 1 in.
56	Channel 1 out.
57	Same as above, Channel 2 in.
60	Channel 2 out.
61	Same as above, Channel 3 in.
62	Channel 3 out.

3. Decimal/Edit Escape Set

Following are the optional Decimal/Edit Escape Set which is available for the 9120, 9130, 9140, and 9150 processors.

<u>Function Code</u>	<u>Operation</u>
53	Unpack, the contents of 44 contiguous memory locations are changed from zoned to Decimal format.
54	The contents of 22 contiguous memory locations are changed from decimal to zoned format.
55	Edit, the contents of 44 contiguous memory locations are edited according to a corresponding mask.
56	Add, the contents of three contiguous memory locations in decimal format are added to three contiguous memory locations. Both operands are 17 digits plus sign and form a 17 digit sum with sign.
57	Subtract, the contents of three contiguous memory locations in decimal format are subtracted from the contents of three memory locations. Both operands are 17 digits plus sign and form a difference of 17 digits with sign.
60	Multiply, the contents of three contiguous memory locations in decimal format are multiplied by three memory locations. Both operands are 17 digits plus sign forming a 34 digit product with sign.
61	Divide, the contents of six contiguous memory locations are divided by the contents of three contiguous memory locations forming a 17 digit signed quotient, and a 17 digit remainder.
62	Logical (absolute) Compare, from one to 24, eight bit characters in memory (operand 1) are compared to a corresponding operand 2, setting appropriate skip designators.
63	Decimal Skip on Y mask.

Where: 2^0 = Plus
 2^1 = Negative
 2^2 = Zero
 2^3 = Overflow

4. Fixed and Floating Point

The CONUS arithmetic units will be free standing cabinets which will communicate with the CONUS over a special 24 or 48 bit channel. The unit will include all the arithmetic algorithms for the CONUS. Data will be transferred to, and operated on, in the arithmetic unit with special instructions in the CONUS Escape Mode. The arithmetic unit will contain a 24 or 48 bit program addressable Accumulator, and a 24 or 48 bit program addressable quotient register. All arithmetic and logical operations will be performed in 2's complement binary. The unit will contain the necessary circuitry for translation and execution of the instructions which will be transferred to the unit from the processor. The standard set of instructions are listed below and are self explanatory.

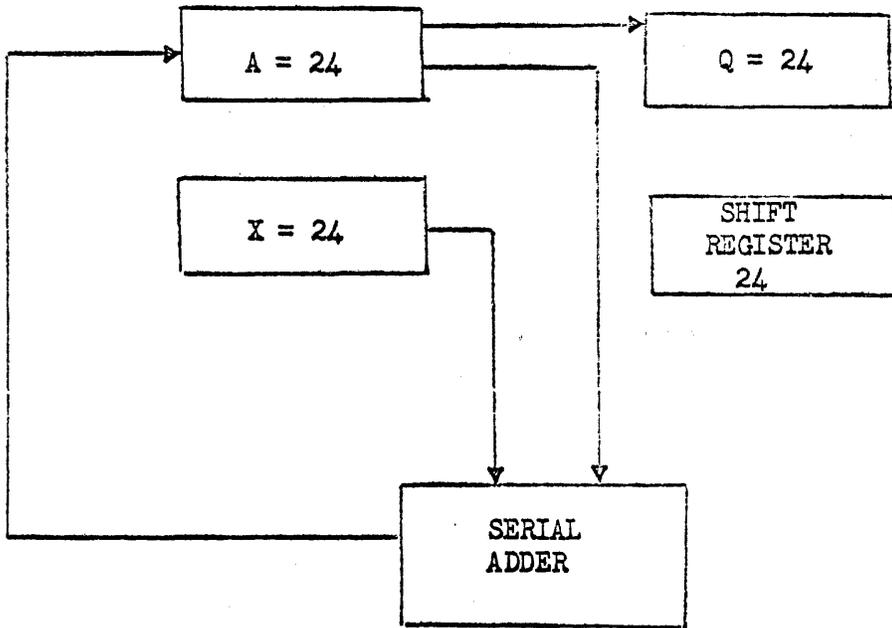
There will be a total of 6 completely compatible arithmetic units which will be available as an option - three 24 bit and three 48 bit units.

The floating point adapter will be available only with the three 48 bit arithmetic units. The units are called out as follows:

FX241	Slow speed 24 bits
FX242	Medium speed 24 bits
FX243	High speed 24 bits
FX481	Slow speed 48 bits
FX482	Medium speed 48 bits
FX483	High speed 48 bits

53	ADD	$A + (Y) \rightarrow A$
54	SUBTRACT	$A - (Y) \rightarrow A$
55	MULTIPLY	$A \times (Y) \rightarrow AQ$
56	DIVIDE	$AQ \div (Y) \rightarrow A$ remainder in Q
57	LOAD A	$(Y) \rightarrow A$
60	LOAD Q	$(Y) \rightarrow Q$
61	STORE A	$A \rightarrow (Y)$
62	STORE Q	$Q \rightarrow (Y)$
620	SHIFT LEFT A	A left Y places
621	SHIFT RIGHT A	A right Y places
622	SHIFT LEFT Q	Q left Y places
623	SHIFT RIGHT Q	Q right Y places
624	SHIFT LEFT AQ	AQ left Y places
625	SHIFT RIGHT AQ	AQ right Y places
626	FLOATING UNPACK	} 48 Set Only
627	SET BASE	
63	FLOATING ADD	
64	FLOATING SUBTRACT	
65	FLOATING MULTIPLY	
66	FLOATING DIVIDE	
67	SKIP CONDITIONAL	
70	SET BASE	
71	LOGICAL PRODUCT	$A \odot (Y) \rightarrow A$
72	LOGICAL SUM	$A \oplus (Y) \rightarrow A$
73	LOGICAL DIFFERENCE	$A \ominus (Y) \rightarrow A$
74	SELECTIVE SUB	

FX241

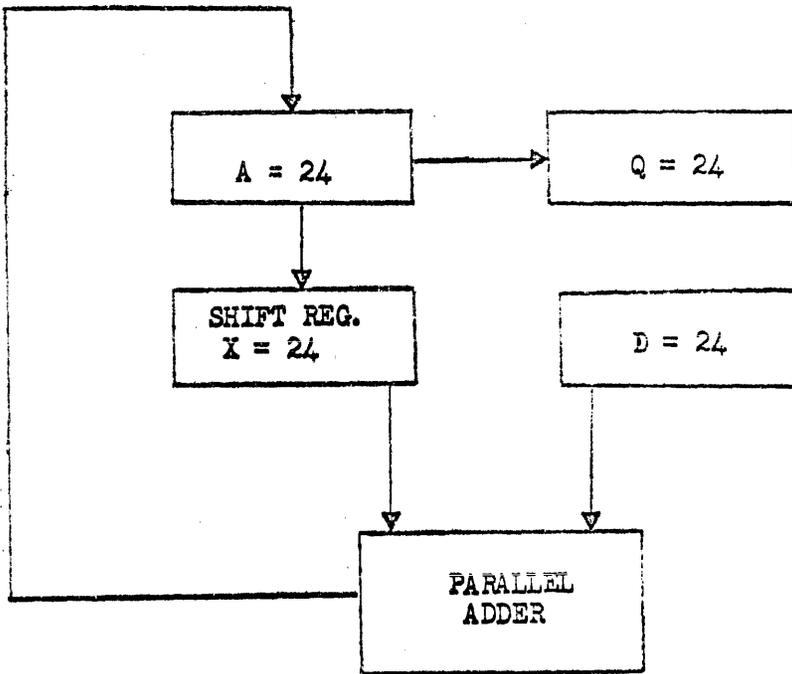


Add Time \approx 11 uSec.

Multiply Time \approx 144 uSec. Avg.

Divide Time \approx 278 uSec.

EX242

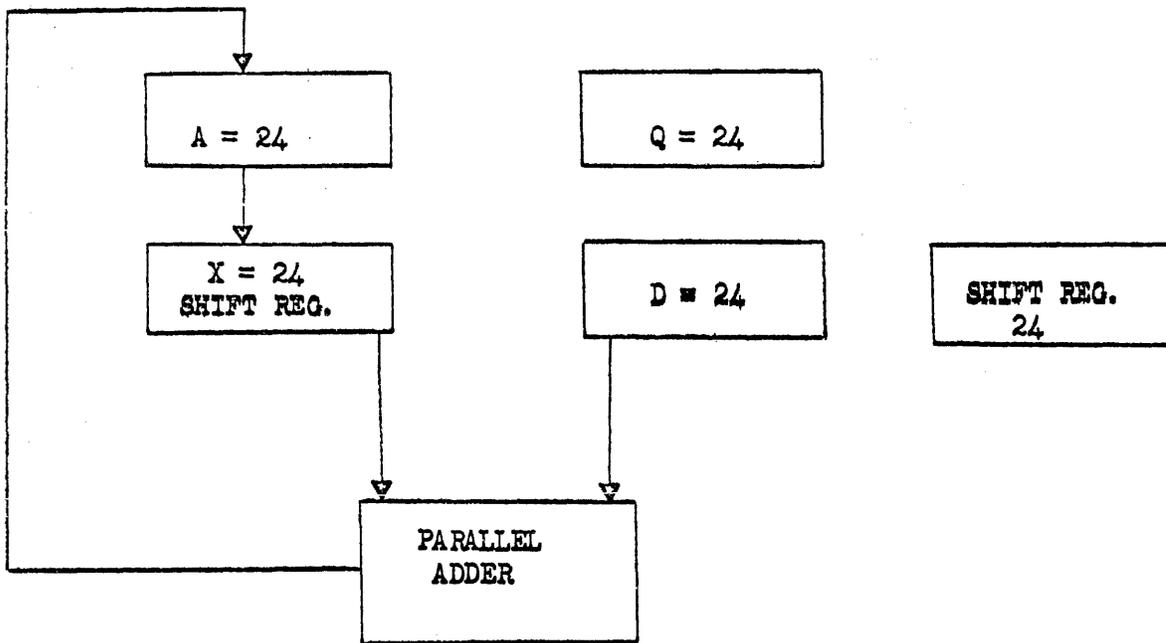


Add Time \approx .2 uSec.

Multiply Time \approx 12 uSec. Avg.

Divide Time \approx 15 uSec.

FX243

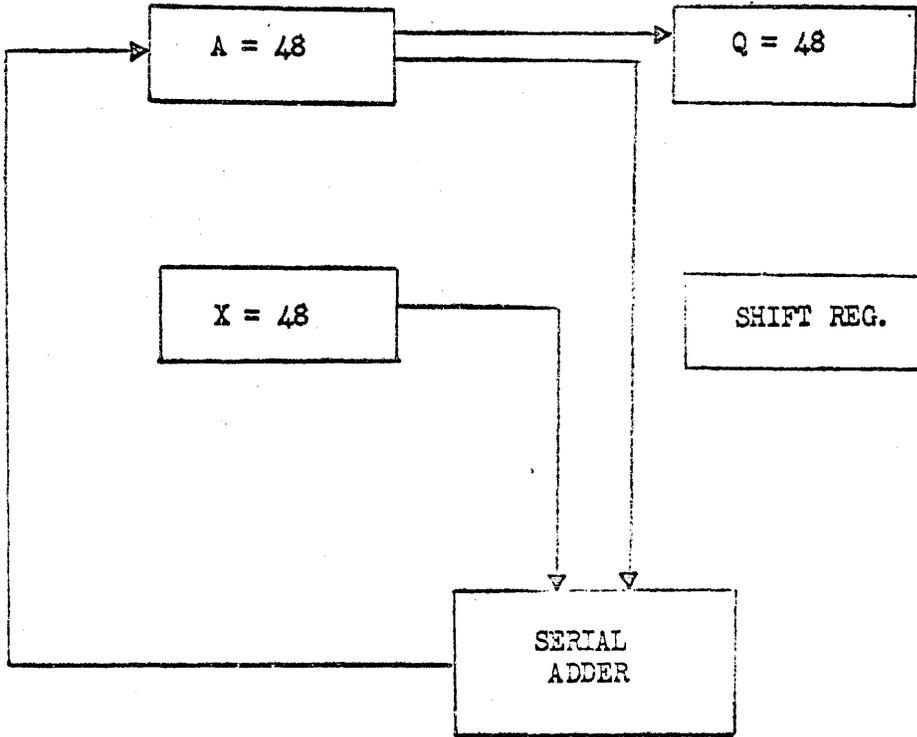


Add Time \approx .2 uSec.

Multiply Time \approx 3.6 uSec. Avg.

Divide Time \approx 4.8 uSec.

FX481

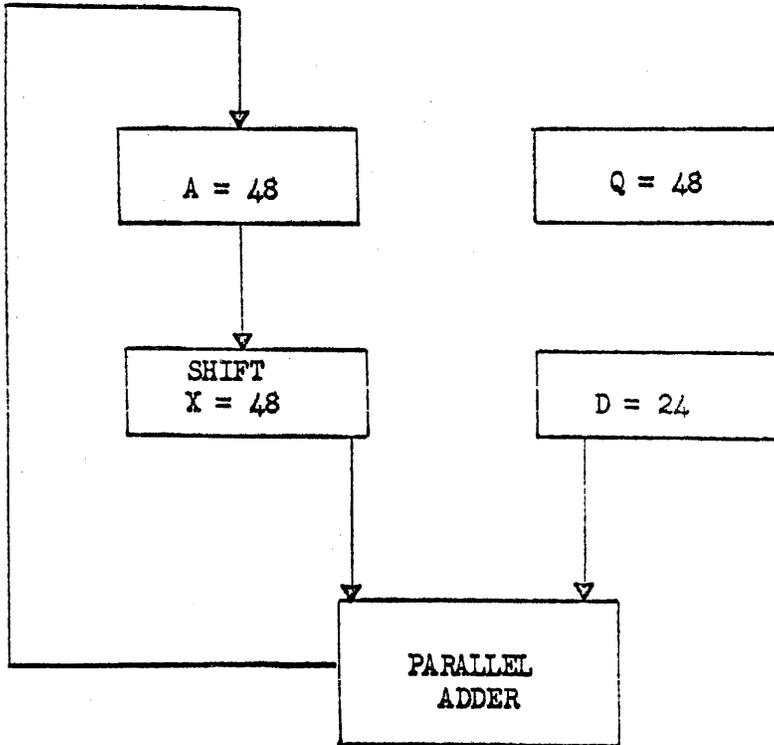


Add Time \approx 21 μ Sec.

Multiply Time \approx 525 μ Sec. Avg.

Divide Time \approx 1030 μ Sec.

FX482

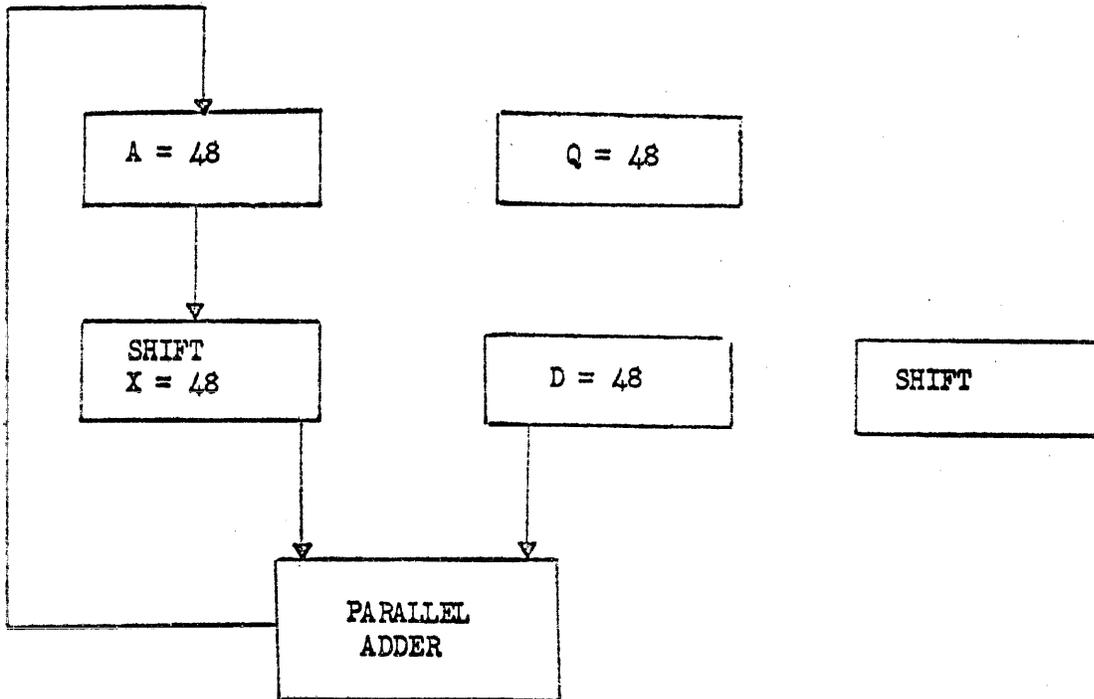


Add Time \approx .2 uSec.

Multiply Time \approx 24 uSec. Avg.

Divide Time \approx 29 uSec.

FX480

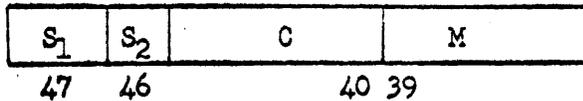


Add Time \approx .1 uSec.

Multiply Time \approx 2.4 uSec.

Divide Time \approx 2.6 uSec.

There will be 2 floating point adapters available - FLP1 and FLP2* with the 48 bit arithmetic units. FLP1 will automatically normalize all arithmetic results to the base 2. FLP2 will automatically normalize all arithmetic results to a variable base (2, 4, 16, 256) according to a 3 bit "base register" (1, 2, 4, 8) as set by the programmer with a special instruction. The floating point format will be as follows:



Where:

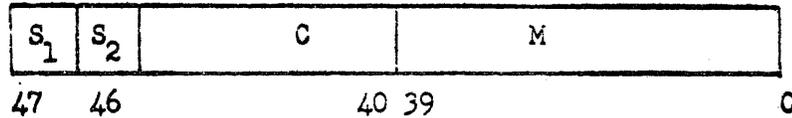
M (Mantissa) = 40 bits
C (Characteristic) = 6 bits
S₁ (Sign of Mantissa) = 1 bit
S₂ (Sign of Characteristic) = 1 bit

Floating algorithms will have approximately the same execution time as fixed point. To these arithmetic units will be added an ultra high speed 48 bit unit for competing with the CDC 6600 and IBM 92.

*See Enclosure

CONUS 24 Floating Point

Assuming a 48 bit optional arithmetic section, the following floating point package will be applicable:



Where:

- M (Mantissa) = 40 bits
- C (Characteristic) = 6 bits
- S_1 (Sign of Mantissa) = 1 bit
- S_2 (Sign of Characteristic) = 1 bit

The basic algorithm for floating point is identical to that used presently in the 1107 with the exception of renormalization. The renormalization portion of the Floating Point algorithm will be variable according to a variable characteristic base (2, 4, 16, 256) as designated by the programmer, i.e., renormalize the mantissa in groups of 1, 2, 4, 8, respectively.

If X is the mantissa and § is the base, X is normalized if the following holds:

$$1/\S \leq X < 1$$

Given the floating point format as described above, the following numerical ranges (N) are possible:

- base 2 $2^{-63} \leq N \leq 2^{63} \approx 10^{-21} \leq N \leq 10^{21}$
- base 4 $2^{-126} \leq N \leq 2^{126} \approx 10^{-42} \leq N \leq 10^{42}$
- base 16 $2^{-252} \leq N \leq 2^{252} \approx 10^{-84} \leq N \leq 10^{84}$
- base 256 $2^{-504} \leq N \leq 2^{504} \approx 10^{-168} \leq N \leq 10^{168}$

Implementation of the variable base will be accomplished by a 3 bit (binary) base register which may be set by the programmer with a special instruction. The register will be used by the hardware to determine the appropriate scaling procedure at the completion of each floating point arithmetic instruction. The cost of incorporating this feature is in the neighborhood of \$300 to \$400.

The feature is justified from both an applications and a marketing standpoint. From an application standpoint there are two basic advantages of variable base floating point over fixed base floating point. 1) Increased significance when dealing with small numbers. 2) Ability to handle large numbers, and obviously there are no disadvantages, in that the identical results can be achieved by using the same base. From the marketing standpoint we need only say, "We offer everything that the IBM 360 offers (floating point to the base 16) and more."

APPENDIX B

<u>FUNCTION</u>	<u>OPERATION</u>	<u>DESIGNATOR</u>	<u>TIME</u>
00	Stop	Clear	4.8
01	Store Prog. Status Code	Clear	7.2
02	Load Prog. Status Code	Fill	7.2
03	Execute Remote Y	(Variable)	7.2+
04	Execute Remote Y Set Return	(Variable)	7.2+
05	Skip on Y Mask	Hold	4.8 - NS 7.2 - S
06	Jump to Y	Hold	4.8
07	Set/Clear Lockout (Y $2^0=1$ Set, $2^0=0$ Clear)		4.8
10	Add (R) + (Y) \rightarrow R	Fill	9.6
11	Subt. (R) - (Y) \rightarrow R	Fill	9.6
12	AND (R) \odot (Y) \rightarrow R	Fill	9.6
13	EX OR (R) \oplus (Y) \rightarrow R	Fill	9.6
14	STORE (B) _B at Y	Hold	9.6
15	Load B _B from Y	Hold	9.6
16	B + Const. (B) _B + Y \rightarrow B _B	Hold	7.2
17	Compare B (B) _B - (Y) B _B + 1 \rightarrow B	Fill	12
20	Load R with (Y)	Fill	9.6
21	Load R with (Y) _U	Fill	9.6
22	Load R with (Y) _M	Fill	9.6
23	Load R with (Y) _L	Fill	9.6
24	Compare (R) - (Y)	Fill	9.6
25	Compare (R) - (Y) _U	Fill	9.6
26	Compare (R) - (Y) _M	Fill	9.6
27	Compare (R) - (Y) _L	Fill	9.6
30	Store (R) at Y	Hold	9.6
31	Store (R) _L at Y _U	Hold	9.6

APPENDIX B

<u>FUNCTION</u>	<u>OPERATION</u>	<u>DESIGNATOR</u>	<u>TIME</u>
32	Store $(R)_L$ at Y_M	Hold	9.6
33	Store $(R)_L$ at Y_L	Hold	9.6
34	Inc. (Y)	Fill	7.2
35	Ent. R with Const. + Sign	Fill	7.2
36	Dec. (Y) - 1, Skip if Y_f Neg.	Hold	7.2 - NS 9.6 - S
37	Store Address	Hold	9.6
40	Shift Storage L.C. 8	Fill	7.2
41	Shift R right K(0-7)	Fill	4.8+2.4 (K)
42	Shift R left K(0-7)	Fill	4.8+2.4 (K)
43	Parity on (Y) Des. (Set Carry if Even, Plus 0 and Pos.)	Fill	7.2
44	Reserved		
45	Reserved		
46	Reserved		
47	Reserved		
50	Reserved		
51	Reserved		
52	Escape to Mode Y	Hold	4.8
53	Activate Input Chan. Y_N (Mask) $2^0 = \text{Chan}_0$, Etc.	Hold	4.8
54	Activate Output Chan. Y_N (Mask)	Hold	4.8
55	EF R \rightarrow Chan. Y_N (Translate Y)	Hold	7.2
56	Store ESI Chan. Y_N , Ack. Y_N (Translate Y)	Hold	7.2

Note:

Function Codes 57 through 77 are defined within their respective Escape class. Function Codes 53 through 56, above, are in the Executive Control Escape class. There are shown here to indicate the complete Instruction Repertoire for the 9110 processor.