NCR REFERENCE MANUAL

An Educational Publication

number: 13
page: 1 of 21
date: Oct. 73

ST-9402-09
BINDER NO. 0141

NCR CENTURY 251 AND 300 COMMAND TIMING

This publication, intended as a supplement to the "NCR Century 251 Processor," "NCR Century 300 Processor", and "NCR Century 251 and 300 Hardware Commands" publications, explains the methods for determining command timing for the NCR Century 251 and 300 Processors. It is divided into three major sections; (1) NCR Century 251 Command Timing, (2) NCR Century 300 Command Timing, and (3) NCR Century 251 and 300 Command Execution Time Equations. The command timing sections explain the methods for determining command setup time and the manner in which it is related to command execution time; the execution time equations section contains the related execution time equation for each command.

NCR CENTURY 251 COMMAND TIMING

Command timing for the NCR Century 251 is determined by the command setup and execution times. The command setup and execution times are, in turn, determined by the size of the command (single- or double-stage), the mode or modes of addressing used with the command, and the particular command for which the timing is desired. The command timing equation is CT = TS + TE, where TS is the setup time and TE is the execution time. The approximate amount of time necessary to perform a series of commands is determined by adding together the command times of each command in the series.

251 Command Setup Time

Command setup time is determined by the size of the command being setup and the mode or modes of addressing used during the setup operation. The times shown in the following tables vary depending upon index register usage and command size. The first table "BASIC SETUP TIMES", includes the times necessary to perform mode 0 addressing and index register manipulation for mode 0, 1, 2, or 3 addressing; these values must be used in addition to the mode 1, 2, or 3 addressing values contained in the "COMPLEX SETUP TIME" table, when other than mode 0 addressing is used with a command. The times indicated in both tables are approximate times based on a memory cable length of 20 feet, and assume that all addresses are 0 mod 4 (evenly divisible by four) so that only one read cycle is necessary to read a word (four bytes of data).

BASIC SETUP TIME

Command	Time
Single Stage	
without index register	1.99 μsec
with index register	3.50 μsec
Double Stage	
without index registers	3.98 μsec
with index register for one stage	5.49 μsec
with index registers for both stages	7.00 μsec

The following table contains the complex setup times for mode 1, 2, and 3 addressing; the times are added to the basic setup time when these modes of addressing are used with a command. Since a double-stage command may specify more than one mode of addressing, the time necessary to perform complex addressing should be calculated separately for each stage and then added to the basic setup time.

COMPLEX SETUP TIME

Addressing Mode	Time
Mode 1 (once for each level)*	
No index register specified in field accessed	1.51 μsec
Index register specified in field accessed	3.02 μsec
Mode 2	
3 bytes of accessed field in one word	1.75 μsec
3 bytes of accessed field spread across two words	3.02 μsec
Mode 3	1.42 μsec

* If terminating addressing with mode 0, time for it is included in mode 1 time; if mode 2 or 3 is used for termination, add appropriate time as specified in table.

EXAMPLE:

Assume that a double-stage command is set up with one stage using an index register with mode 0 addressing, and the other stage using two levels of mode 1 addressing and terminating with mode 2 addressing; both levels of mode 1 addressing use an index register, and the mode 2 addressing, which does not use an index register, accesses a field with 3 bytes in one word.

7.00 μsec	Basic setup time -	Includes index register setup for both mode 0 and first level of mode 1 addressing.
3.02 μsec	Complex setup time -	Includes final operation of first level of mode 1 addressing and initial index register setup for second level of mode 1 addressing.
1.51 μsec	Complex setup time -	Includes final operation of second level of mode 1 addressing.
1.75 μsec	Complex setup time -	Mode 2 addressing (3 bytes in word).
13.28 μsec	Total setup time	

251 Command Execution Time

Command execution time is determined according to the specific command for which the time is desired as well as by the length of the fields on which the command operates. For this reason execution time varies between commands and differs for the same command during various operations. The execution times may be calculated from the execution time equations contained in the NCR Century 251 and 300 Command Execution Time Equations section of this publication; the execution time, once calculated for a given command and specific use, may then be added to the setup time for that command to determine the total command time.

NCR CENTURY 300 COMMAND TIMING

The NCR Century 300 processor sets up and executes commands concurrently; that is, as one command is being executed by the E-unit, another one is being either partially or completely set up by the I-unit. In some instances the result of the command being executed may effect the setup operation of the command which is currently being setup; if this happens, an I-unit abort occurs, terminating the setup operation. The processor then stops concurrent operation and begins sequential operation. During sequential operation the command setup that was aborted is reinitiated after the command which caused the abort completes execution. After the reinitiated setup is completed, the processor resumes concurrent operation, and executes the aborted command while setting up the next command.

To determine the command timing for the NCR Century 300, the command setup and execution times as well as the amount of time during which operations overlap must be taken into consideration. The general command timing equation is CT = TS + TE - TO, where TS is setup time, TE is execution time, and TO is the time during which the setup of the current command overlaps the execution of the command that was previously setup. Command timing for one command on the NCR Century 300, then, is actually a function of two commands; the command for which the time is desired and the command that is performed prior to it. The approximate time necessary to complete a series of operations may be determined by using the command timing equation to establish the command time for each command in a series and adding the resulting times together.

300 Command Setup Time

Command setup time for the NCR Century 300 is determined by the size of the command (single- or double-stage), the modes of addressing used with each stage of the command, and by processor operation in either concurrent mode, or due to an I-unit abort, in sequential mode.

- Concurrent Operation

  Command setup during concurrent operation consists of basic setup when mode 0 addressing is performed, or basic and complex setup when mode 1, 2, or 3 addressing is performed. During basic setup, the I-unit accesses the command, determines the size and modes of addressing, and then calculates the intermediate addresses for both stages, using indexing when specified. Basic setup takes place while the E-unit is executing the command that was previously set up. If mode 0 is the only mode of addressing specified in the command, I-unit operation terminates on completion of basic setup and the E-unit executes the current command after execution of the previous command is completed.

  When mode 1, 2, or 3 addressing is specified by the command currently being setup, the I-unit performs basic setup, calculating intermediate addresses, and then stops operation until the E-unit completes execution of the previous command. After the E-unit has completed execution of the previous command, the I-unit performs complex setup where the final addresses from Modes 1 and 2 addressing are calculated, and where incrementing is performed for mode 3 addressing. When complex setup is completed, the E-unit executes the command while the I-unit performs basic setup for the next command.

  The following table contains the approximate times required for the basic setup of a command; if only one memory storage unit is available to the processor, the times labeled "TIME WITH CONTENTION" should be used in calculating setup time. If more than one memory storage unit is available to the processor, the times labeled "TIME WITHOUT CONTENTION" apply. It is assumed that all memory cables are 20 feet in length and that all addresses are 0 mod 4 (evenly divisible by four). If the previously executed command was a successful BRANCH command, .240 μsec must be added to the basic setup time, and if a command is being repeated by the repeat flow, .480 μsec must be added.

BASIC SETUP TIME

Command	Time w/o Contention	Time with Contention
Single Stage		
without index register	1.68 μsec	1.79 μsec
with index register	2.88 μsec	3.10 μsec
Double Stage		
without index registers	3.12 μsec	3.34 μsec
with index register for one stage	4.32 μsec	4.65 μsec
with index registers for both stages	5.52 μsec	5.96 μsec

The following table contains the complex setup times for mode 1, 2, and 3 addressing; the times are added to the basic setup time when the modes of addressing are specified for a command. Since double-stage commands may specify more than one mode of addressing, the time necessary to perform complex setup should be determined separately for each stage of a command and then added to the basic setup time.

COMPLEX SETUP TIME

Addressing Mode	Time w/o Contention	Time with Contention
Mode 1 (once for each level of addressing) *		
No index register specified in field accessed	1.44 μsec	1.55 μsec
Index register specified in field accessed	2.64 μsec	2.86 μsec
Mode 2		
3 bytes of accessed field in one word	1.68 μsec	1.79 μsec
3 bytes of accessed field spread across two words	2.64 μsec	2.86 μsec
Mode 3	1.37 μsec	1.48 μsec

* If terminating addressing with mode 0, time for it is included in mode 1 time; if mode 2 or 3 is used for termination, add appropriate time as specified in table.

EXAMPLE:

Assume that a double-stage command is setup with one stage using an index register with mode 0 addressing, and the other stage using two levels of mode 1 addressing and terminating with mode 2 addressing; both levels of mode 1 addressing use an index register and the mode 2 addressing, which does not use an index register, accesses a field with 3 bytes in one word. Memory contention does not occur during the setup.

5.52 μsec	Basic setup time	- Includes index register setup for both mode 0 and first level of mode 1 addressing.
2.64 μsec	Complex setup time	- Includes final operation of first level of mode 1 addressing and initial index register setup for second level of mode 1 addressing.
1.44 μsec	Complex setup time	- Includes final operation of second level of mode 1 addressing.
1.68 μsec	Complex setup time	- Mode 2 addressing (3 bytes in 1 word).
11.28 μsec	Total setup time	

- I-unit Aborts

During concurrent operation, if a condition arises where the continued operation of the I-unit would result in the erroneous setup of a command, an I-unit abort occurs, terminating the setup operation which is currently taking place. The I-unit then reinitiates the setup operation after the E-unit completes the execution of the command that was previously setup, and begins sequential operation.

The following conditions cause an I-unit abort:

1. The E-unit writing in memory locations used by the I-unit. (This can occur only during the basic setup operation with simultaneous operation.)

2. Execution of the COUNT, TESTB, TESTCE, TESTCU, REPEAT (A=0), RESTORE, and WAIT commands (the abort occurs for the TEST commands only when program control is transferred and for the COUNT command only when control is not transferred).

3. Mode 3 addressing of the A operand where both the A and B operands use the same index register (this can only occur during complex setup with a double-stage command).

If an I-unit abort is caused by either the E-unit writing in memory locations used by the I-unit or by a transfer of control command (COUNT, TESTB, etc.), the setup time for the command is established during the sequential setup operation, which is explained later; if mode 3 addressing causes an I-unit abort, however, setup time is the sum of the I-unit abort time and sequential setup time. The values in the following table may be used to calculate I-unit abort time. Since the abort condition is detected during complex setup of mode 3 addressing, I-unit abort time includes basic setup of the A and B operands, any levels of mode 1 addressing associated with the A operand, either with or without index register usage, and mode 3 addressing time of the A operand, up to the time the abort condition is detected.

MODE 3 I-UNIT ABORT TIME

Command	Time w/o Contention	Time with Contention
Basic Setup		
Index register initially assigned to both operands	5.52 µsec	5.96 µsec
Index register initially assigned only to B operand (mode 1 addressing preceding mode 3 addressing for A)	4.32 µsec	4.65 µsec
Mode 1 (once for each level is mode 1 addressing for A, if any, prior to mode 3 addressing)		
No index register specified in field being accessed	1.44 µsec	1.55 µsec
Index register specified in field being accessed	2.64 µsec	2.86 µsec
Mode 3	1.37 µsec	1.48 µsec

- Sequential Operation

Sequential operation is performed when an I-unit abort occurs. If the I-unit abort was caused by either the E-unit writing in memory locations used by the I-unit or due to the execution of a transfer command (COUNT, TESTB, etc.), the setup time is only that time used for sequential setup; if the I-unit abort was caused by mode 3 addressing of the A operand, however, the setup time is equal to the sum of the I-unit abort and sequential setup times. The following table contains the approximate times required to calculate the sequential setup time for a single-stage command; if the setup time for a double-stage command is desired, use the values in the following table to calculate the setup time for each stage separately and then add the resulting times together.

SEQUENTIAL SETUP TIME

Addressing Mode	Time w/o Contention	Time with Contention
Basic Setup or Mode 0 addressing *		
Without index register	1.44 µsec	1.55 µsec
With index register	2.64 µsec	2.86 µsec
Mode 1 (once for each level of addressing) **		
No index register specified in field accessed	1.20 µsec	1.31 µsec
Index register specified in field accessed	2.40 µsec	2.62 µsec
Mode 2		
3 bytes in one word	1.44 µsec	1.55 µsec
Three bytes spread across two words	2.40 µsec	2.62 µsec
Mode 3	1.13 µsec	1.24 µsec

* Add .24 µsec to the total sequential setup time determined for the command; this time is required by the I-unit transfer flow.

** If terminating with mode 0, time for it is included in mode 1 time; if mode 2 or 3 is used for termination, add appropriate time as specified in table.

300 Command Execution Time

Command execution time is determined according to the specific command for which the time is desired as well as by the length of the fields on which the command operates. For this reason, execution time varies between commands and differs for the same command during various operations. The execution times may be calculated from the execution time equations contained in the NCR Century 251 and 300 Command Execution Time Equations section of this publication; the execution time, once calculated for a given command and specific use, is then added to the setup time before setup and execution overlap time is calculated.

300 Setup and Execution Overlap Time

After setup and execution time has been determined for a given command, the overlap time is calculated. The overlap time is that time during which the execution of the command setup previously overlaps the basic setup time of the command currently being setup; the overlap time, once determined, is then subtracted from the sum of the setup and execution times for the desired command.

The following table contains the criteria for calculating overlap time. The variables in the table are E1 and TBS2, where E1 represents the execution time of the command previously setup and TBS2 represents the basic setup time of the command for which the timing is desired (the command currently being setup).

OVERLAP TIME

Condition	Over Lap Time (To)
Concurrent operation or sequential operation due to mode 3 abort	
E1 >TBS2	TBS2
E1 <TBS2	E1
Transfer or E-unit writing in I-unit memory abort	0

The following illustration is a graphic representation of the setup and execution times, and their relationship to each other. Four commands, in various combinations, are used in this example:

1. MVAL (Move A Left to Right)
2. BADD (Binary Add)
3. MVAR (Move A Right to Left)
4. LDMONR (Load Monitor Register)

The times used in the illustration are defined below:

TE = Execution time
TBS - Basic setup time
TCI = Complex setup time
TSS = Sequential setup time
TEI = I-unit command execution time

The column "COMMAND CHARACTERISTICS" describes the type of command being executed and the setup stages that the I-unit goes through. The column "SYMBOLIC CHARACTERISTIC" is an abbreviated symbol version of the first column. The column "UNIT" contains the letters "I" and "E" indicating the unit to which the graphic representation refers. The column "RELATIVE TIMING" contains the graphic time representation of the I- and E-unit operations and their relationship to each other. The numbers enclosed in parentheses refer to the commands listed above.

RELATIVE COMMAND SETUP AND EXECUTION TIMING

COMMAND CHARACTERISTICS	SYMBOLIC CHARACTERISTICS	UNIT	RELATIVE TIMING
NORMAL CONDITIONS			
Basic setup time is greater than the execution time. I-unit does not enter the complex setup stage.	TBS > TE	I	TBS(1) TBS(2) TBS(3)
	No TCI	E	TE(1) TE(2)
Basic setup time is less than the execution time. I-unit does not enter the complex setup stage.	TBS < TE	I	TBS(1) TBS(2) TBS(3)
	No TCI	E	TE(1) TE(2)
Basic setup time is greater than the execution time. I-unit enters the complex setup stage.	TBS > TE	I	TBS(1) TBS(2) TCI(2) TBS(3)
	TCI	E	TE(1) TE(2)
Basic setup time is less than the execution time. I-unit enters the complex setup stage.	TBS < TE	I	TBS(1) TBS(2) TCI(2) TBS(3)
	TCI	E	TE(1) TE(2)
I-UNIT EXECUTION REQUIRED			
Basic setup time is greater than the execution time. i—unit does not enter the complex setup stage.	TBS > TE	I	TBS(1) TBS(4) TEI(4) TBS(3)'
	No TCI	E	TE(1)
Basic setup time is less than the execution time. I-unit does not enter the complex setup stage.	TBS < TE	I	TBS(1) TBS(4) TEI(4) TBS(3)
	No TCI	E	TE(1)
Basic setup time is greater than the execution time. I-unit enters the complex setup stage.	TBS > TE	I	TBS(1) TBS(4) TCI(4) TEI(4) TBS(3)
	TCI	E	TE(1)
Basic setup time is less than the execution time. I-unit enters the complex setup stage.	TBS < TE	I	TBS(1) TBS(4) TCI(4) TEI(4) TBS(3)
	TCI	E	TE(1)
ABORT CONDITIONS			
Abort occurs during the basic setup stage.	Abort TBS	I	TBS(1) TBS(2) ABORT TSS(2) TBS(3)
		E	TE(1) TE(2)
Abort occurs during the complex setup stage (Mode 3 abort).	Abort TCI	I	TBS(1) TBS(2) TCI(2) TSS(2) TBS(3)
		E	TE(1) ABORT TE(2)

NCR CENTURY 251 AND 300 COMMAND EXECUTION TIME EQUATIONS

The command execution time equations consists of the description of the terms that are used in the timing equations and the explicit equations for calculating the execution time of each hardware command.

Description of Terms in Timing Equations

AV = the average value of the digits in the multiplier or the quotient.

A_{21} = lower two bits of the A address.

A_2A_1 = 0, if A is 0 mod 4.
= 1, if A is 1 mod 4.
= 2, if A is 2 mod 4.
= 3, if A is 3 mod 4.

BCT = between commands testing.

B_{21} = lower two bits of the B address.

B_2B_1 = same as for A_2A_1, except for the B field.

EA = exponent in the A field of floating point commands.

EB = exponent in the B field of floating point commands.

E_{21} = lower two address bits of the last character scanned in the SCAN commands (Command Codes 55, 56, and 57).

FPADD = 1 for commands 74, 75, 78, and 79, if the A and B fields have like signs.

= 1 for commands 76, 77, 7A, and 7B, if the A and B fields have unlike signs.

if (FPADD) = 1, (FPADD) ' = 0

N = a number that is greater than or equal to one, or smaller than or equal to T ($1 \leq N \leq T$), which is the first character in the field, specified by the T character, with $b_8 = 1$; used in the DECODE commands (Command Codes 4E and 4F).

NB = any number represented in binary or hexadecimal.

NN = number of digits to be shifted to normalize the A or B operands, before floating point multiplication or division.

NNA = number of digits to be shifted to normalize the A operand, before floating point multiplication or division.

NNB = number of digits to be shifted to normalize the B operand, before floating point multiplication or division.

NNS = number of digits to be shifted left to normalize the results of floating point addition or subtraction.

(OF = 1) = exponent overflow occurs when the results of floating point addition or subtraction are shifted in order to normalize the results.

P = machine cycle, approximately 240 nanoseconds for both the NCR Century 251 and 300 processors.

R = read cycle time which does not overlap the P time, approximately 670 + 3L nanoseconds for the NCR Century 251 and 660 + 3L nanoseconds for the NCR Century 300. L is equal to memory cable length in feet. *

T = number in the T character.

TA = value of T character bits 5 to 8.

TB = value of T character bits 1 to 4.

TN = number of bits ON in positions 1 through 6 of the T character of the SHIFT BINARY command (Command Code 35). If bit 2 is ON, add 1 to TN.

T/2 = T divided by 2 with any fraction unequal to zero raised to the next integer (if $T \neq 0 \bmod 4$, T/4 becomes T/4 + 1; if $T = 0 \bmod 4$, T/4 remains T/4).

T/4 = T divided by 4, with any fraction unequal to zero raised to the next integer (if $T \neq 0 \bmod 4$, T/4 becomes T/4 + 1; if $T = 0 \bmod 4$, T/4 remains T/4).

T_8 = 8th bit of the T character.

(UF = 1) = exponent underflow occurs when the results of floating point addition or subtraction are shifted in order to normalize the results.

W = write cycle time which does not overlap the P time, approximately 340 + 3L nanoseconds for the NCR Century 251 and 350 + 3L nanoseconds for the NCR Century 300. L is equal to memory cable length in feet. *

* These times do not take into consideration any interference at the memory ports from other units requesting access to the same module. If interference occurs, an additional 110 nanoseconds is required for each read or write access for the NCR Century 300 and an additional 300 nanoseconds for each read and write access for the NCR Century 251; these times apply only when the system configuration has one memory storage unit, Interference can be effectively ignored on NCR Century 300 systems with more than one memory storage unit.

WA = number of times the A field is accessed in memory. **

WB = number of times the B field is accessed in memory. **

WBC = number of words read from the B field, until the T character is decremented to zero or a comparison is made; used in the SCAN commands (Command Codes 55, 56, and 57).

(WBM = 1) = number of times a word in the B field is accessed in which at least one byte has $b_7 = 1$.

(WBM = 0) = number of times a word in the B field is accessed in which all bytes have $b_7 = 0$.

(X) = contents of X.

1 mod 4 = any number equal to 0 mod 4 plus 1.

2 mod 4 = any number equal to 0 mod 4 plus 2.

3 mod 4 = any number equal to 0 mod 4 plus 3.

( )' = logical NOT of quantity.

+ = logical OR of two values when used in the description of timing equations.

+ = arithmetic add when used in the timing equations.

) . (= logical AND of two values when used in the description of timing equations.

|| = absolute value of any quantity.

** The number of memory accesses (WA or WB) is determined on a word basis, as follows:

WA or WB = T/4 or T/4 + 1, or T/4 + 2.

The number of memory accesses are dependent on the starting address of the operands and the T character.

Execution Time Equations

- Fixed Point Binary Commands

C = 51 (33) (B3) WORD COMPARE BINARY

TE = 6P + 2R

C = 53 (35) (B5) SHIFT BINARY

TE = 5P + R + W   single word shift = 0
TE = (5 + TN)P + R + W   single word
TE + 7P + 2R + 2W   double word shift = 0
TE = (7 + 2TN)P + 2R + 2W   double word
add 1 P if (T8 = 1) • (T7 = 1)

C = 54 (36) (B6) MULTIPLY BINARY

TE = 24P + 2R + 2W

C = 55 (37) (B7) DIVIDE BINARY

TE = 43P + 3R + 2W

C = 66 (42) (C2) WORD ADD BINARY

C = 67 (43) (C3) WORD SUBTRACT BINARY

TE = 5P + 2R + W

C = 96 (60) (E0) ADD BINARY

C = 97 (61) (E1) SUBTRACT BINARY

TE = 5P (3P + 2R + W)WB
add (1P + 1R) if $A_{21} < B_{21}$

C = 101 (65) (E5) COMPARE BINARY

TE = 6P (3P + 2R)WB
add (1P + 1R) if $A_{21} < B_{21}$

C = 114 (72) (F2) BINARY TO DECIMAL CONVERSION

TE = 105P + 1R + 2W   Minimum if binary number $<989,680_{16}$
TE = 113P + 1R + 2W   Maximum if binary number $989,680_{16} \leq N_B \leq FFF,FFF_{16}$
if $N_B > FFF,FFF_{16}$ use value between minimum and maximum values.

C = 115 (73) (F3) DECIMAL TO BINARY CONVERSION

TE = 43P + 2R + W
add 1P if number is negative

- Floating Point Commands

C = 116 (74) (F4) FLOATING POINT ADD SINGLE

C = 118 (76) (F6) FLOATING POINT SUBTRACT SINGLE

$TE = 9P + 2R + W$ if $(FPADD) \cdot (EA = EB) \cdot (NNS = 0,7) \cdot$
$(UF)'$ or $(FPADD) \cdot (EA = EB) \cdot$ (underflow)

add 1P if $EA > EB$

add 2P if $(EA = EB) \cdot (FPADD)'$

add 2P if $(EA \neq EB)$

add 1P if $(EA \cdot EB)$ is odd

add 1P if $(EA \neq EB) \cdot (FPADD)'$

add 2P if $(FPADD)'$ and result is negative

add 2P if $(NNS = 1, 2, 4, 6) \cdot (UF)'$

add 4P if $(NNS = 3,5) \cdot (UF)'$

add 2P if $(NNS = 1, 2, 4, 6) \cdot (UF)$

add 2P if $(NNS = 3, 5) \cdot (UF)$ with underflow on first shift of 1, 2, 4 characters

add 4P if $(NNS = 3, 5) \cdot (UF)$ with underflow on second shift of 1 character.

C = 117 (75) (F5) FLOATING POINT ADD DOUBLE

C = 119 (77) (F7) FLOATING POINT SUBTRACT DOUBLE

$TE = 13P + 4R + 2W$ if $(EA = EB) \cdot (FPADD) \cdot (NNS = 0, 14) \cdot (UF)$

add 1P if $EA > EB$

add 2P if $(EA = EB) \cdot (FPADD)'$

add 2P if $|EA - EB| = 2, 4, 6$

add 4P if $|EA - EB| = 1, 3, 5, 7, 8, 10, 12, 14, \ldots$

add 6P if $|EA - EB| = 9, 11, 13, 15, 17, \ldots$

add 2P if $(FPADD)'$

add 4P if $(FPADD)'$ and result is negative

add 1P if $NNS = -1$

add 2P if $(NNS = 1, 2, 4, 6, 8) \cdot (UF)'$

add 4P if $(NNS = 3, 5, 7, 9, 10, 12) \cdot (UF)'$

add 6P if $(NNS = 11, 13) \cdot (UF)'$

add 4P if underflow occurs with first shift of 1, 2, 4, 6, 8 digits for NNS = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13

add 6P if underflow occurs with second shift of 1, 2, 4, 6, 8 digits for NNS = 3, 5, 7, 9, 10, 11, 12, 13

add 8P is underflow occurs with last shift of 1 digit for NNS = 11, 13

C = 120 (78) (F8) FLOATING POINT MULTIPLY AND ADD SINGLE

$TE = (TE_M + TE_A) - 1P$

TE_M is the execution time for the multiply part of Multiply and Add command. (For analysis see FLOATING POINT MULTIPLY SINGLE command.)

TE_A is the execution time for the addition part of Multiply and Add Command. (For analysis see FLOATING POINT ADD SINGLE command.)

C = 121 (79) (F9) FLOATING POINT MULTIPLY AND ADD DOUBLE

$TE = (TE_M + TE_A) - 1P$

TE_M is the execution time for the multiply part of Multiply and Add Double command. (For analysis see FLOATING POINT MULTIPLY DOUBLE command.)

TE_A is the execution time for the addition part of Multiply and Add Double command. (For analysis see FLOATING POINT ADD DOUBLE command.)

C = 122 (7A) (FA) FLOATING POINT COMPARE SINGLE

TE = 8P + 2R if (PFADD) • (EA = EB)

add 1P if EA < EB

add 2P if (EA = EB) • (FPADD)'

add 2P if (EA ≠ EB)

add 1P if |EA - EB| is odd

add 1P if (EA ≠ EB) • (FPADD)'

add 2P if (FPADD)' and result is negative

C = 123 (7B) (FB) FLOATING POINT COMPARE DOUBLE

TE = 11P + 4R if (FPADD) • (EA = EB)

add 1P if EA > EB

add 2P if (EA = EB) • (FPADD)'

add 2P if |EA - EB| = 2, 4, 6

add 4P if |EA - EB| = 1, 3, 5, 7, 8, 10, 12, 14, 16, ...

add 6P if |EA - EB| = 9, 11, 13, 15, 17, ...

add 2P if (FPADD)'

add 4P if (FPADD)' and result is negative

C = 124 (7C) (FC) FLOATING POINT MULTIPLE SINGLE

TE = 9P + 1R + 1W if A operand = 0

TE = 11P + 2R + 1W if (B operand = 0) • (NNA = 0)

TE = 13P + 2R + W if (B operand = 0) • (NNA = 1, 2)

TE = 15P + 2R + W if (B operand = 0) • (NNA = 3, 4, 5)

if A and B are not equal to zero TE = 24P + 2R + W plus:

add 2P if NNA = 1, 2

add 4P if NNA = 3, 4, 5

add 2P if NNB = 1, 2, 4

add 4P if NNB = 3, 5

C = 125 (7D) (FD) FLOATING POINT MULTIPLY DOUBLE

TE = 10P + 2R + 2W if A operand = 0
TE = 12P + 4R + 2W if (B operand = 0) • (NNA = 0)
TE = 14P + 4R + 2W if (B operand = 0) • (NNA = 1, 2)
TE = 16P + 4R + 2W if (B operand = 0) • (NNA = 3, 4, 5, 6, 7, 8, 9, 10)
TE = 18P + 4R + 2W if (B operand = 0) • (NNA = 11, 12, 13)

If A and B are not equal to zero TE = 71P + 4R + 2W plus:
add 2P if NNA = 1, 2
add 4P if NNA = 3, 4, 5, 6, 7, 8, 9, 10
add 6P if NNA = 11, 12, 13
add 2P if NNB = 1, 2, 4
add 4P if NNB = 3, 5, 6, 7, 8, 9, 10, 12
add 6P if NNB = 11, 13

C = 126 (7E) (FE) FLOATING POINT DIVIDE SINGLE

TE = 4P + R if A operand = 0, PE trap is taken
TE = 11P + 2P + W if (B operand = 0) • (NNA = 0)
TE = 13P + 2R + W if (B operand = 0) • (NNA = 1, 2)
TE = 15P + 2R + W if (B operand = 0) • (NNA = 3, 4, 5)

If A and B are not equal to zero TE = 36P + 2R + W plus:
add 2P if NNA = 1, 2
add 4P if NNA = 3, 4, 5
add 2P if NNB = 1, 2, 4
add 4P if NNB = 3, 5
add 4P if dividend < divisor after normalization of both divisor and dividend.

C = 127 (7F) (FF) FLOATING POINT DIVIDE DOUBLE

TE = 4P + 2R if A operand = 0, PE trap is taken
TE = 13P + 4R + 2W if (B operand = 0) • (NNA = 0)
TE = 15P + 4R + 2W if (B operand = 0) • (NNA = 1, 2)
TE = 17P + 4R + 2W if (B operand = 0) • (NNA = 3, 4, 5, 6, 7, 8, 9, 10)
TE = 19P + 4R + 2W if (B operand = 0) • (NNA = 11, 12, 13)

If A and B operands are not equal to zero TE = 64P + 4R + 2W plus:
add 2P if NNA = 1, 2
add 4P if NNA = 3, 4, 5, 6, 7, 8, 9, 10
add 6P if NNA = 11, 12, 13
add 2P if NNB = 1, 2, 4
add 4P if NNB = 3, 5, 6, 7, 8, 9, 10, 12
add 6P if NNB = 11, 13
add 8P if dividend < divisor after normalization of both divisor and dividend.

- Decimal Arithmetic Commands

C = 64 (40) (C0) ADD SIGNED

C = 65 (41) (C1) SUBTRACT SIGNED

TE = 7P + (3P + 2R + W) WB + 2R not recomplement answer
TE = 10P + (6P + 3R + 2W) WB + 2R recomplement answer
add (1P + 1R) if $A_{21} < B_{21}$

C = 69 (45) (C5) COMPARE SIGNED

TE = 8P + (3P + 2R) WB + 2R
add (1P + 1R) if $A_{21} > B_{21}$

C = 93 (5D) (DD) MULTIPLY SIGNED

TE = 26P + 4(AV) (TA - 1/2)P + 2R + W if (TB ≤ 4) · (TA ≤ 4)
TE = 30P + 4(AV) (TA - 1/2)P + 2R + W if (TB ≤ 4) . (TA > 4)
TE = 28P + 8(AV) (TA - 1/2)P + 2R + W if (TB > 4) . (TA ≤ 4)
TE = 32P + 8(AV) (TA - 1/2)P + 2R + W if (TB > 4) . (TA > 4)
add 1R if TA > 4 - A_2A_1
add 1R if TB > 4 - B_2B_1
add 1R if TA > 8 - A_2A_1
add 1R if TB > 8 - B_2B_1
add 1W if TA + TB > 4
add 1W if TA + TB > 8
add 1W if TA + TB > 12
Max value of (AV) is 9 for all 9's in multiplier
Min value of (AV) is 1 for all 1's in multiplier
Most likely value of (AV) is 5

C = 98 (62) (E2) ADD UNSIGNED

C = 99 (63) (E3) SUBTRACT UNSIGNED

TE = 5P + (3P + 2R + W) WB
add (1P + 1R) if $A_{21} < B_{21}$

C = 113 (71) (F1) DIVIDE SIGNED

TE = 25P + 6(TB - TA)P + 4(AV) (TB - TA)P + 2R + 2W if TA ≤ 4
TE = 25P + 8(TB - TA)P + 8(AV) (TB - TA)P + 3R + 3W if TA > 4
where TB > TA
add 1W if (TB - TA) > 5
add 1R if TA > 4 - A_2A_1
add 1R if TA > 8 - A_2A_1
add (1P + 1R) if (TB > 5) · (TA > 4)
add (1P + 1R) if (TB ≥ 13) + (TB ≤ 12) · (TB ≥ 9) · (TA ≤ 4)
Max value of (AV) is 9 for all 9's in quotient
Min value of (AV) is 1 for all 1's in quotient
Most likely value of (AV) is 5.

C = 76 (4C) (CC) PACK

$TE = 3P + T/4\ (4P + W) + (WA)R$
add 1P if $(A = 3 \text{ Mod } 4) + (A = 2 \text{ or } 3 \text{ Mod } 4) \cdot (B = 2 \text{ or } 3 \text{ Mod } 4) +$
$(t \text{ even}) \cdot (B = 2 \text{ or } 3 \text{ Mod } 4) \cdot (A = 1 \text{ Mod } 4)$

C = 77 (4D) (CD) UNPACK

$TE = 3P + T/2\ (4P + W) + (WA)R$
add 3P if $(A = 1 \text{ or } 3 \text{ Mod } 4) + (B = 1 \text{ or } 3 \text{ Mod } 4)$

- Move Data Commands

C = 68 (44) (C4) MOVE B RIGHT TO LEFT

C = 100 (64) (E4) MOVE A RIGHT TO LEFT

$TE = 5P + (2P + R + W)\ WB$
add $(1P + 1R)$ if $A_{21} < B_{21}$

C = 84 (54) (D4) MOVE A LEFT TO RIGHT

$TE = 2P + (2P + R + W)\ WB$
add $(1P + 1R)$ if $A_{21} < B_{21}$

- Logic Commands

C = 73 (49) (C9) EDIT

$TE = P + (3P + W)T + (2P + 1R)\ WA + (P + R)\ ((WBM = 1) + (WBM = 0))$

C = 78 (4E) (CE) DECODE TO DELIMITER

C = 79 (4F) (CF) DECODE ALL

$TE = 5P + (P + R + W)\ WB + (5P + R)\ N + W$
$1 \leq N \leq T$ for DECODE TO DELIMITER
$N = T$ for DECODE ALL

C = 85 (55) (D5) SCAN ON KEY LESS

C = 86 (56) (D6) SCAN ON KEY EQUAL

C = 87 (57) (D7) SCAN ON KEY GREATER

$TE = (2E_2E_1 - 1 - 2B_{21})\ P + (9P + R)WBC + R + W$

C = 92 (5C) (DC) TABLE COMPARE

$TE = 3P + (7P + 3R)\ N + R\ (WB)$
$1 \leq N \leq T$ N depends on whether T character is exhausted or a
Compare occurs first

C = 94 (5E) (DE) LOGIC

TE = 6P + 2R + W if T8 = 0
TE = 11P + 2R + W if T8 = 1

- Transfer Commands

C = 74 (4A) (CA) COUNT

TE = 4P + R + W if Count Register = 0
TE = 5P + R + W if Count Register ≠ 0

C = 75 (4B) (CB) JUMP

TE = 5P + W

C = 81 (51) (D1) TEST CHARACTER EQUAL

C = 82 (52) (D2) TEST CHARACTER UNEQUAL

C = 83 (53) (D3) TEST BIT

TE = 4P + R if not Branch
TE = 5P + R if Branch

C = 104 (68) (E8) BRANCH OVERFLOW

C = 105 (69) (E9) BRANCH LESS

C = 106 (6A) (EA) BRANCH EQUAL

C = 107 (6B) (EB) BRANCH LESS OR EQUAL

C = 108 (6C) (EC) BRANCH GREATER

C = 109 (6D) (ED) BRANCH GREATER OR LESS

C = 110 (6E) (EE) BRANCH GREATER OR EQUAL

C = 111 (6F) (EF) BRANCH UNCONDITIONAL

TE = 3P

- Special Commands

C = 49 (31) (B1) LOAD T

TE = 4P + R

C = 50 (32) (B2) TEST AND SET LOCK

TE = 3P + R (Locked)
TE = 5P + R (Not Locked)

C = 70 (46) (C6) SET IP ON

C = 71 (47) (C7) SET IP OFF

TE = 3P

C = 72 (48) (C8) RESTORE

TE = 5P + 2R

C = 80 (50) (D0) OPTION SWITCHES INPUT

C = 91 (5B) (DB) STORE TRACE

TE = 4P + W

C = 88 (58) (D8) LOAD BAR

TE = 5P + 2R

C = 89 (59) (D9) LOAD MONITOR REGISTER

TE = 0 No E-unit flow, I - unit execution = 2P + R

C = 90 (5A) (DA) LOAD TRACE

TE = 3P + R

C = 102 (66) (E6) REPEAT

TE = 4P + R + W if (A) ≠ 0
TE = 5P + 2R + W if (A) = 0 and next command is single stage
TE = 6P + 2R + W if (A) = 0 and next command is double stage

REPEAT FLOW add to every command when RI is on

TE = 4P + R + W if (32) = 0
TE = 5P + R + W if (32) ≠ 0

C = 112 (70) (F0) INOUT

with Q8 = 1
TE = 5P + R

with Q8 = 0
busy response
TE = 11P + R + W

not busy
TE = 13P + R + W
add 3P for each additional PAF character sent to peripheral

add (2P + R) for each time must access four new PAF characters

C = 103 (67) (E7) WAIT

TE = 195P for each cycle from BCT to BCT

C = 36 (24) (A4) MOVE EFFECTIVE B

TE = 5P + W, (A = 0 mod 4)
TE = 4P + W, (A = 1 mod 4)
TE = 6P + 2W, (A = 2 mod 4, or A = 3 mod 4)