DATE: May 12, 1978 PE-T-416

TO: All R & D Personnel

FROM: M. Sporer

SUBJECT: P400 Instructions Times

This document lists the instruction times for the P400 64V-mode non-restricted instructions. The times listed are the best-case times determined from an analysis of the microcode. This list has been made to ease the task of comparing the relative performance of 64V-mode instructions. Many assembly-language programmers have alternate codings for a particular algorithm and this list can help assess the relative performance of the alternate methods. The actual instruction times will depend on factors other than the micro-code specified time. Some of these are:

- the number of cache references
- the cache hit rate
- the amount and type of memory
- the STLB hit rate
- the process exchange rate

With these factors in mind, the list can be used to determine which instructions are relatively faster than others. Since the actual instruction times have many contingencies, the speed of various instructions may be affected in different ways under different circumstances. Despite these variations, the list can be used to compare instructions which will be executed in the same environment.

The type column indicates the format and/or function of the operation.

AP:	Three-word operation, the last two words of which are an AP address pointer.
BR:	Two-word operation, the second word of which is a word number within the current procedure segment to which to branch.
CON:	Single-word control operation.
FLD:	Single-word field operation.
FOPR:	Single-word floating-point operation.
FSKP:	Single-word floating-point skip operation.
LOG:	Single-word logicize operation.
MODE:	Single-word mode operation.
MR:	Memory reference operation.
OPR:	Single-word miscellaneous operation.
SH:	Single-word shift operation.
SKP:	Single-word skip operation.

The C column indicates the effect of the operation on the C-bit and the L-bit.

-:	C and L are unchanged by the operation.
1:	C is unchanged, L is carry.
2:	C is overflow, L is carry.
3:	C is overflow, L is indeterminant.
4:	C is shift extension, L is indeterminant.
5:	C is a result of the operation, L is indeterminant.
6:	C and L are indeterminant.
7:	C and L are loaded by the operation.
8:	C is cleared, L is indeterminant.

The CC column indicates the effect of the operation on the condition codes.

-:	Condition codes are unchanged by the operation.
1,4:	Condition codes indicate the result of the arithmetic operation or compare.
5:	Condition codes are indeterminant.
6:	Condition codes are loaded by the operation.
7:	Condition codes indicate the result of the operation.

The time column indicates the time in nanoseconds that the operation takes. The special symbols are interpreted as follows:

*	add 520 for 32-bit indirection
*1	add 280 for long-form add 240 for 16-bit indirection
*2	add 1000 for indirection as required
*3	2nd time if right character
*4	2nd time if value in A or L is <0
*5	2nd time if increment or decrement reaches 0
*6	depends on result of compare :

compare	>	=	<
CAZ	680	760	800
CAS	760	840	880
CLS	1240	1320	1360
FCS	2660	3220	2740
DFCS	2460	3770	2810

*7	2nd time if item added or removed from queue
*8	2nd time if store is performed
*9	2nd time if branch taken or logicize result is 1 or skip taken
n	number of shifts or normalizes needed
op	instruction to be executed
a/n	either adjust or normalize but not both
a	number of arguments

mnem	opcode	time(nsec)		type	C	cc	description
A1A	141206	480		OPR	2	1	Add 1 to A Register. A+1=>A.
A2A	140304	480		OPR	2	1	Add 2 to A Register. A+2=>A.
ABQ	141716	3560-4640	*7	AP	-	6	Add to Bottom of Queue. CCEQ = FULL
ACA	141216	480		OPR	2	1	Add CBIT to A Reg. CBIT+A=>A.
ADD	06	560	*1	MR	2	1	Add. (A) + [EA]16 => A.
ADL	06 03	1120	*	MR	2	1	Add long (no hole). (A,B) + [EA]32 => A,B.
ADLL	141000	680		OPR	2	1	Add Link to L Register.
ALFA	001301	2840	*2	FLD	6	5	Add L to Field Address Register Zero.
ALFA	001311	2520	*2	FLD	6	5	Add L to Field Address Register One.
ALL	0414XX	560+160n		SH	4	5	A Left Logical.
ALR	0416XX	560+160n		SH	4	5	A Left Rotate.
ALS	0415XX	640+160n		SH	4	5	A Left Shift (Short Integer Arithmetic).
ANA	03	560	*1	MR	-	-	And. (A) .AND. [EA]16 => A.
ANL	03 03	1080	*	MR	-	-	And long. (A,B) .AND. [EA]32 => A,B.
ARGT	000605	3080+2400a	*	CON	-	-	Argument transfer.
ARL	0404XX	560+160n		SH	4	5	A right logical.
ARR	0406XX	560+160n		SH	4	5	A right Rotate.
ARS	0405XX	560+160n		SH	4	5	A right shift (short integer).
ATQ	141717	3600- 4520	*7	AP	-	6	Add to top of Queue. CCEQ = full.
BCEQ	141602	600- 960	*9	BR	-	-	Branch on Condition Code .EQ.
BCGE	141605	600- 960	*9	BR	-	-	Branch on Condition Code .GE.
BCGT	141601	600- 960	*9	BR	-	-	Branch on Condition Code .GT.
BCLE	141600	600- 960	*9	BR	-	-	Branch on Condition Code .LE.
BCLT	141704	840- 1200	*9	BR	-	-	Branch on Condition Code .LT.
BCNE	141603	600- 960	*9	BR	-	-	Branch on Condition Code .NE.
BCR	141705	840- 1200	*9	BR	-	-	Branch on C-bit Reset.
BCS	141604	600- 960	*9	BR	-	-	Branch on C-bit Set.
BDX	140734	760- 880	*9	BR	-	-	Branch on Decremented X.
BDY	140724	760- 880	*9	BR	-	-	Branch on Decremented Y.
BEQ	140612	600- 960	*9	BR	-	4	Branch on A Register .EQ. 0.
BFEQ	141612	600- 960	*9	BR	-	4	Branch on Floating Accumulator .EQ. 0.
BFGE	141615	600- 960	*9	BR	-	4	Branch on Floating Accumulator .GE. 0.
BFGT	141611	600- 960	*9	BR	-	4	Branch on Floating Accumulator .GT. 0.
BFLE	141610	600- 960	*9	BR	-	4	Branch on Floating Accumulator .LE. 0.
BFLT	141614	600- 960	*9	BR	-	4	Branch on Floating Accumulator .LT. 0.
BFNE	141613	600- 960	*9	BR	-	4	Branch on Floating Accumulator .NE. 0.
BGE	140615	600- 960	*9	BR	-	4	Branch on A Register .GE. 0.
BGT	140611	600- 960	*9	BR	-	4	Branch on A Register .GT. 0.
BIX	141334	760- 880	*9	BR	-	-	Branch on Incrementec X.
BIY	141324	760- 880	*9	BR	-	-	Branch on Incrementec Y.
BLE	140610	600- 960	*9	BR	-	4	Branch on A Register .LE. 0.

mnem	opcode	time(nsec)	type	C	cc	description
BLEQ	140702	600- 960 *9	BR	-	4	Branch on L Register .EQ. 0.
BLGE	140615	600- 960 *9	BR	-	4	Branch on L Register .GE. 0.
BLGT	140701	600- 960 *9	BR	-	4	Branch on L Register .GT. 0.
BLLE	140700	600- 960 *9	BR	-	4	Branch on L Register .LE. 0.
BLLT	140614	600- 960 *9	BR	-	4	Branch on L Register .LT. 0.
BLNE	140703	600- 960 *9	BR	-	4	Branch on L Register .NE. 0.
BLR	141707	840- 1200 *9	BR	-	-	Branch on Link Reset.
BLS	141706	840- 1200 *9	BR	-	-	Branch on Link Set.
BLT	140614	600- 960 *9	BR	-	4	Branch on A Register .LT. 0.
BMEQ	141602	600- 960 *9	BR	-	-	Branch on Magnitude-Concitions L,CC .EQ.
BMGE	141706	840- 1200 *9	BR	-	-	Branch on Magnitude-Concitions L,CC .GE.
BMGT	141710	600- 1280 *9	BR	-	-	Branch on Magnitude-Concitions L,CC .GT.
BMLE	141711	880- 1240 *9	BR	-	-	Branch on Magnitude-Concitions L,CC .LE.
BMLT	141707	840- 1200 *9	BR	-	-	Branch on Magnitude-Concitions L,CC .LT.
BMNE	141603	600- 960 *9	BR	-	-	Branch on Magnitude-Conditions L,CC .NE.
BNE	140613	600- 960 *9	BR	-	4	Branch on A Register .NE. 0.
CAL	141050	480	OPR	-	-	Clear A Reg. Left Byte.
CAR	141044	480	OPR	-	-	Clear A Reg. Right Byte.
CAS	11	760- 880 *6	MR	1	1	Compare. Skip 0,1,2 instructions if (A) >,=,< [EA]16.
CAZ	140214	680- 800 *6	OPR	1	1	Compare A with 0. Skip 0,1,2 INST. if A >,=,< 0
CGT	001314	1520	-	6	5	Computed Go To.
CHS	140024	480	OPR	-	-	Change Sign of A Register.
CLS	11 03	1240- 1360 *6	MR	1	1	Compare long (no hole). Skip 0,1,2 instructions if (A,B) >,=,< [EA]32. [EA]32.
CMA	140401	480	OPR	-	-	One'S Complement A Register.
CRA	140040	480	OPR	-	-	Clear A Register. 0=>A.
CRB	140015	640	OPR	-	-	Clear B Register. 0=>B.
CRE	141404	480	OPR	-	-	Clear E. 0=>E.
CRL	140010	480	OPR	-	-	Clear L Register. 0=>L.
CRLE	141410	720	OPR	-	-	Clear L and E. 0=>L, 0=>E.
CSA	140320	640	OPR	5	-	Copy Sign of A. A1=>CBIT, 0=>A1.
DFAD	06 02	4420+160a/n*	MR	3	5	Double floating add. (DFAC + [EA]64 => DFAC.
DFCM	140574	2640	FOPR	3	5	Double Floating Complement. -DFAC=>DFAC.
DFCS	11 02	2460- 3770 *6	MR	6	5	Double floating compare. Skip 0,1,2 instructions if (DFAC) >,=,< [EA]64.
DFDV	17 02	22960 *	MR	3	5	Double floating civice. (DFAC) / [EA]64 => DFAC
DFLD	02 02	1640 *	MR	-	-	Double floating load. [EA]64 => DFAC.

mnem	opcode	time(nsec)		type	C	cc	description
DFLX	15 02	960	*	MR	-	-	Load double floating index. 4*[EA]16 => X.
DFMP	16 02	19060	*	MR	3	5	Double floating multiply. (DFAC) * [EA]64 => DFAC.
DFSB	C7 02	4420+160a/n	*	MR	3	5	Double floating subtract. (DFAC) - [EA]64 => DFAC
DFST	04 02	3200	*	MR	-	-	Double floating store. (DFAC) => [EA]64.
DIV	17	5040	*1	MR	3	5	Divide (with hole). (A,B) / [EA]16 => A; remainder => B.
DIV	17	4760	*1	MR	3	5	Divide (integer, no hole). (A,B) / [EA]16 => A; remainder => B.
DRX	140210	560- 800	*5	OPR	-	-	Decremented and Skip if 0.
DVL	17 03	5940	*	MR	3	5	Divide long (integer, no hole). (A,B;EH,EL) / [EA]32 => A,B; remainder EH, EL.
E16S	000011	1280		MODE	-	-	Enter P300 16K Sectored Mode.
E32R	001013	1280		MODE	-	-	Enter P300 32K Relative Mode.
E32S	000013	1280		MODE	-	-	Enter P300 32K Sectored Mode.
E64R	001011	1280		MODE	-	-	Enter P300 64K Relative Mode.
E64V	000010	1280		MODE	-	-	Enter P400 Mode.
EAA	01 01	840	*	MR	-	-	Effective address to A-register. EA => A.
EAFA	001300	2440	*2	AP	-	-	Effictive Address to Field Register Zero.
EAFA	001310	2440	*2	AP	-	-	Effective Address to Field Register One
EAL	01 01	840	*	MR	-	-	Load effective address. EA => A,B.
EALB	13 02	800	*	MR	-	-	Effective address to linkage base register EA => LB.
EAXB	12 02	800	*	MR	-	-	Effective address to temporary base register. EA => XB.
ERA	05	560	*1	MR	-	-	Exclusive OR. (A) .XOR. [EA]16 A.
ERL	05 03	1080	*	MR	-	-	Exclusive OR long. (A,B) .XOR. [EA]32 => A,B.
FAD	06 01	3500+160a/n	*	MR	3	5	Floating add. (FAC) + [EA]32 => FAC.
FCM	140530	1880		FOPR	3	5	Floating COMPLEMENT. -FAC=>FAC.
FCS	11 01	2660- 3220	*6	MR	6	5	Floating compare. Skip 0,1,2 instructions if (FAC) >,=,< [EA]32.
FDBL	140016	640		FOPR	-	-	Convert SLE FLOAT to DOUBLE FLOAT FAC=>DFAC.
FDV	17 01	11200	*	MR	3	5	Floating divide. (FAC) / [EA]32 => FAC.
FLD	C2 01	2040	*	MR	-	-	Floating load. [EA]32 => FAC.
FLTA	140532	2800+160n		FOPR	3	5	Convert Integer to Floating. FLOT(A)=>FAC.
FLTL	140535	2800+160n		FOPR	8	5	Convert Long Integer to Floating FLOT(L)=>FAC.

mnem	opcode	time(nsec)		type	C	cc	description
FLX	15 01	840	*	MR	-	-	Load floating index. 2*[EA]16 => X.
FMP	16 01	8280	*	MR	3	5	Floating multiply. (FAC) * [EA]32 => FAC.
FRN	140534	2520		FOPR	3	5	Floating ROUND UP.
FSB	07 01	3500+160a/n	*	MR	3	5	Floating subtract. (FAC) - [EA]32 => FAC.
FSGT	140515	1200- 1440	*9	FSKP	-	4	Floating Skip if .GT. 0.
FSLE	140514	1200- 1440	*9	FSKP	-	4	Floating Skip if .LE. 0.
FSMI	140512	1200- 1440	*9	FSKP	-	4	Floating Skip if .LT. 0.
FSNZ	140511	1200- 1440	*9	FSKP	-	4	Floating Skip if .NE. 0.
FSPL	140513	1200- 1440	*9	FSKP	-	4	Floating Skip if .GE. 0.
FST	04 01	1840	*	MR	6	6	Floating store. (FAC) => [EA]32.
FSZE	140510	1200- 1440	*9	FSKP	-	4	Floating Skip if .EQ. 0.
IAB	000201	880		OPR	-	-	Interchange A Reg. with B Reg. A=B & B=>A.
ICA	141340	480		OPR	-	-	Interchange Bytes of A Reg.
ICL	141140	460		OPR	-	-	Interchange Bytes of A Reg. and Clear Left
ICR	141240	480		OPR	-	-	Interchange Bytes of A Reg. and Clear Right
ILE	141414	1000		OPR	-	-	Interchange L and E. L=>E & E=>L.
IMA	13	960	*1	MR	-	-	Interchange memory and A-register.
INTA	140531	3000+160n		FOPR	3	5	Convert Floating to Integer. INT(FAC)=>A.
INTL	140533	2840+160n		FOPR	3	5	Convert Floating to Long Integer. INT(FAC)=>L.
IRS	12	560- 720	*5 *1	MR	-	-	Increment, replace, and skip if zero.
IRX	140114	560- 800	*5	OPR	-	-	Increment X and Skip if 0.
JMP	01	440	*1	MR	-	-	Unconditional jump. EA => PB,P.
JST	10	1000	*1	MR	-	-	Jump and store. (P) => [EA]16, EA+1 => P.
JSX	35 03	1080	*	MR	-	-	Jump and set X-register. (P) => X, EA => P.
JSXB	14 02	1000	*	MR	-	-	Jump and set temporary base register. (PB,P) => XB, EA, => PB,P.
JSY	14	800	*1	MR	-	-	Jump and set Y-register. (P) => Y, EA => P.
LCEQ	141503	640- 880	*9	LOG	-	-	Logicize on Cond Code EQ. if CC.EQ.,1=>A else 0 ->A.
LCGE	141504	640- 880	*9	LOG	-	-	Logicize on Conc Coce GE. if CC.GE.,1=>A else 0=>A.
LCGT	141505	640- 880	*9	LOG	-	-	Logicize on Cond Coce GT. if CC.GT.,1=>A else 0=>A.
LCLE	141501	640- 880	*9	LOG	-	-	Logicize on Cond Coce LE. if CC.LE.,1=>A else 0=>A.
LCLT	141500	640- 880	*9	LOG	-	-	Logicize on Cond Code LT. if CC.LT.,1=>A else 0=>A.

mnem	opcode	time(nsec)	type	C	cc	description
LCNE	141502	640- 880	*9 LOG	-	-	Logicize on Cond Code NE. if CC.NE.,1=>A else 0=>A.
LDA	02	560	*1 MR	-	-	Load A-register. [EA]16 => A.
LDC	001302	2280- 2480	*3 FLD	-	7	Loac Char to A Reg. as Specifiex by Field Address Register Zero.
LDC	001312	2280- 2480	*3 FLD	-	7	Loac Char to A-Reg. as Specifiec by Field Address Register One.
LDL	02 03	1080	* MR	-	-	Loac long. [EA]32 => A,B.
LDX	35	560	*1 MR	-	-	Loac X-register. [EA]16 => X.
LDY	35 01	840	* MR	-	-	Loac Y-register. [EA]16 => Y.
LEQ	140413	680- 920	*9 LOG	-	4	Logicize on A Reg. EQ. if A.EQ.0, 1=>A else 0=>A.
LF	140416	680	LOG	-	5	Logicize FALSE. 0=>A.
LFEQ	141113	680- 920	*9 LOG	-	4	Logicize on Floating EQ. if FAC.EQ.0, 1=>A else 0=>A.
LFGE	141114	680- 920	*9 LOG	-	4	Logicize on Floating GE. if FAC.GE.0, 1=>A else 0=>A.
LFGT	141115	680- 920	*9 LOG	-	4	Logicize on Floating GT. if FAC.GT.0, 1=>A else 0=>A.
LFLE	141111	680- 920	*9 LOG	-	4	Logicize on Floating LE. if FAC.LE.0, 1=>A else 0=>A.
LFLI	001303	1240	*2 -	-	-	Loac Field Length Register IMMEDIATE Zero.
LFLI	001313	1240	*2 -	-	-	Loao Fielc Length Register IMMEDIATE One.
LFLT	141110	680- 920	*9 LOG	-	4	Logicize on Floating LT. if FAC.LT.0, 1=>A else 0=>A.
LFNE	141112	680- 920	*9 LOG	-	4	Logicize on Floating NE. if FAC.NE.0, 1=>A else 0=>A.
LGE	140414	680- 920	*9 LOG	-	4	Logicize on A Reg. GE. if A.GE.0, 1=>A else 0=>A.
LGT	140415	680- 920	*9 LOG	-	4	Logicize on A Reg. GT. if A.GT.0, 1=>A else 0=>A.
LLE	140411	680- 920	*9 LOG	-	4	Logicize on A Reg. LE. if A.LE.0, 1=>A else 0=>A.
LLEQ	141513	680- 920	*9 LOG	-	4	Logicize on L Reg. EQ. if L.EQ.0, 1=>A else 0=>A.
LLGE	140414	680- 920	*9 LOG	-	4	Logicize on L Reg. GE. if L.GE.0, 1=>A else 0=>A.
LLGT	141515	680- 920	*9 LOG	-	4	Logicize on L Reg. GT. if L.GT.0, 1=>A else 0=>A.
LLL	0410XX	560+160n	SH	4	5	Long Left Logical.
LLLE	141511	680- 920	*9 LOG	-	4	Logicize on L Reg. LE. if L.LE.0, 1=>A else 0=>A.
LLLT	140410	680- 920	*9 LOG	-	4	Logicize on L Reg. LT. if L.LT.0, 1=>A else 0=>A.
LLNE	141512	680- 920	*9 LOG	-	4	Logicize on L Reg. NE. if L.NE.0, 1=>A else 0=>A.
LLR	0412XX	560+160n	SH	4	5	Long Left Rotate.
LLS	0411XX	640+160n	SH	4	5	Long Left Shift (Long Integer Arithmetic in 64V Mode, else DP Integer with Hole).
LLT	140410	680- 920	*9 LOG	-	4	Logicize on A Reg. LT. if A.LT.0, 1=>A else 0=>A.

mnem	opcode	time(nsec)		type	C	cc	description
LNE	140412	680- 920	*9	LOG	-	4	Logicize on A Reg. NE. if A.NE.0, 1=>A else 0=>A.
LRL	0400XX	560+160n		SH	4	5	Long Right Logical.
LRR	0402XX	560+160n		SH	4	5	Long Right Rotate.
LRS	0401XX	640+160n		SH	4	5	Long Right Shift (Long Integer Arithmetic in 64V Mode, else DP with Hole).
LT	140417	680		LOG	-	5	Logicize TRUE. 1=>A.
MIA	12 01	960+	*	MR	-	-	Microcode entrance. May be microprogrammed to be restricted.
MIB	13 01	960+	*	MR	-	-	Microcode entrance. May be microprogrammed to be restricted.
MPL	16 03	8480	*	MR	-	1	Multiply long (integer, no hole). (A,B) * [EA]32 => A,B,EH,EL.
MPY	16	4480	*1	MR	3	1	Multiply (with hole). (A) * [EA]16 => A,B.
MPY	16	4200	*1	MR	-	1	Multiply (integer, no hole). (A) * [EA]16 => A,B.
NOP	000001	720		OPR	-	-	No Operation.
ORA	03 02	840	*	MR	-	-	Or. (A) .OR. [EA]16 => A.
PCL	10 02	12600+2400a	*	MR	7	6	P-400 procedure call.
PID	000211	760- 1000	*4	OPR	-	-	Convert Short Integer to DP Integer with Hole.
PIDA	000115	760		OPR	-	-	Short Integer to Long Integer Conversion A=>L.
PIDL	000305	1040		OPR	-	-	Convert Long Integer to 64 BIT Integer.
PIH	000205	680		OPR	-	-	Convert DP Integer with Hole to Short Integer.
PIHA	000015	1480- 1760	*4	OPR	3	5	Long Integer to Short Integer Conversion. L=>A. IEX on precision loss.
PIHL	000301	1440- 1720	*4	OPR	3	5	Convert 64 BIT Integer to Long Integer (L,E)=>L
PRTN	000611	4400		CON	7	6	Procedure Return.
RBQ	141715	2560- 4320	*7	AP	-	6	Remove from Bottom of Queue. on empty, A=0 and CC'S set EQ.
RCB	140200	400		OPR	5	-	Reset C-bit. 0=>CBIT.
RTQ	141714	2400- 3800	*7	AP	-	6	Remove from top of Queue. on empty, A=0 and CC'S set EQ.
S1A	140110	480		OPR	2	1	Subtract 1 from A Register. A-2=>A.
S2A	140310	480		OPR	2	1	Subtract 2 from A Register. A-2=>A.
SAR	10026X	720- 800	*9	SKP	-	-	Skip if A Reg bit N Reset.
SAS	101260	720- 800	*9	SKP	-	-	Skip if A Reg bit N Set.
SBL	07 03	1120	*	MR	2	1	Subtract long (no hole). (A,B) - [EA]32 => A,B.
SCB	140600	400		OPR	5	-	Set C-bit. 1=>CBIT.
SGT	100220	720- 800	*9	SKP	-	-	Skip if A Reg. .GT. 0.
SKP	100000	480		SKP	-	-	Skip one word.

mnem	opcode	time(nsec)			type	C	cc	description
SLE	101220	720-	800	*9	SKP	-	-	Skip if A Reg. .LE. 0.
SLN	101100	720-	800	*9	SKP	-	-	Skip if A Reg. bit 16 set.
SLZ	100100	720-	800	*9	SKP	-	-	Skip if A Reg. bit 16 .EQ. 0.
SMCR	100200	720-	800	*9	SKP	-	-	Skip if machine check reset.
SMCS	101200	720-	800	*9	SKP	-	-	Skip if machine check set.
SMI	101400	720-	800	*9	SKP	-	-	Skip if A Reg. .LT. 0.
SNZ	101040	720-	800	*9	SKP	-	-	Skip if A Reg. NE. 0.
SPL	100400	720-	800	*9	SKP	-	-	Skip if A Reg. .GE. 0.
SRC	100001	720-	800	*9	SKP	-	-	Skip if C-bit reset.
SSC	101001	720-	800	*9	SKP	-	-	Skip if C-bit set.
SSM	140500	480			OPR	-	-	Set Sign of A Reg. Minus. 1=>A1.
SSP	140100	480			OPR	-	-	Set Sign of A Reg. Plus. 0=>ABIT1.
STA	04	560		*1	MR	-	-	Store A-register. (A) => [EA]16.
STAC	001200	1560-	1800	*8	AP	-	7	Store A Conditional on B=[EA16]. CCEG= Success
STC	001322	2720-	2960	*3	FLD	-	7	Store Char from A Reg. as Specified by Field Address Register Zero.
STC	001332	2720-	2960	*3	FLD	-	7	Store Char from A Reg. as Specified by Field Address Register One.
STEX	001315	2640			OPR	6	5	Stack Extend. L Reg. Has Extent.
STFA	001320	2640		*	AP	-	-	Store Field Address Register Zero
STFA	001330	2640		*	AP	-	-	Store Field Address Register One.
STL	04 03	1600		*	MR	-	-	Store long. (A,B) => [EA]32.
STLC	001204	1600-	1920	*8	AP	-	7	Store L Concitional on E=[EA32]. CCEG= Success
STX	15	560		*	MR	-	-	Store X-register. (X) => [EA]16.
STY	35 02	840		*	MR	-	-	Store Y-register. (Y) => [EA]16.
SUB	07	560		*1	MR	2	1	Subtract. (A) - [EA]16 => A.
SZE	100040	720-	800	*9	SKP	-	-	Skip if A Reg. .EQ. 0.
TAB	140314	520			OPR	-	-	Transfer A to B Reg. A=>B.
TAK	001015	920			OPR	7	6	Transfer A to KEYS.
TAX	140504	640			OPR	-	-	Transfer A Reg. to X Reg. A=>X.
TAY	140505	640			OPR	-	-	Transfer A Reg. to Y Reg. A=>Y.
TBA	140604	520			OPR	-	-	Transfer B Reg. to A Reg. B=>A.
TCA	140407	720			OPR	2	1	Two's Complement A Register. -A=>A.
TCL	141210	760			OPR	2	1	Two's complement L. -L=>L.
TFLL	001323	1400			FLD	-	-	Transfer Field Length Reg. to L. Reg. Zero
TFLL	001333	1400			FLD	-	-	Transfer Field Length Reg. to L Reg. One.
TKA	001005	840			OPR	-	-	Transfer KEYS to A.
TLFL	001321	1720			FLD	-	-	Transfer L to Field Length

mnem	opcode	time(nsec)		type	C	cc	description
							Register Zero.
TLFL	001331	2040		FLD	-	-	Transfer L to Field Length Register One.
TSTQ	141757	2040	*	AP	-	6	Test Queue. A Set to # Items in Queue. CC's.
TXA	141034	640		OPR	-	-	Transfer X Reg. to A Reg. X=>A.
TYA	141124	640		OPR	-	-	Transfer Y Reg. to A Reg. Y=>A.
WCS	0016XX	480		-	-	-	WCS Entrances. UII on no WCS or WCS not Loaded. May be Microprogrammed to be Restricted.
XCA	140104	760		OPR	-	-	Interchange and Clear A. A=>B, 0=>A.
XCB	140204	760		OPR	-	-	Interchange and Clear B Reg. B=>A, 0=>B.
XEC	01 02	1520+op	*	MR	-	-	Execute instruction at effective address. Not all instructions can be executed.