IBM System/34

Reference Summary

ASSEMBLER MACHINE INSTRUCTION SET

Instruction	Op Code	Hex Code	Operand Format Type	Time in Microseconds
Zero and add zoned decimal	ZAZ	x4	1	T2 + 1.2L2 + 0.6(L1-L2) + T3
Add zoned decimal	AZ	x6	1	T2 + 1.8L2 + 1.2(L1-L2) + T3
Subtract zoned decimal	SZ	x7	1	T2 + 1.8L2 + 1.2(L1-L2) + T3
Move hex character	MVX	x8	2	T2 + 1.8
Move characters	MVC	xC	2	T2 + 1.2L
Compare logical characters	CLC	xD	2	T2 + 1.4L
Add logical characters	ALC	xE	2	T2 + 1.8L
Subtract logical characters	SLC	xF	2	T2 + 1.8L
Insert and test characters (last byte significant digit) (last byte not significant digit)	ITC	xB	2	 T2 + 1.2L1 + 0.2 T2 + 1.2L1 + 0.6
Edit (last byte hex 20) (last byte not hex 20)	ED	xA	2	 T2 + 1.2L2 + 0.6L1 T2 + 1.2L2 + 0.6L1 + 0.8
Move logical immediate	MVI	xC	3	T1 + 0.6
Compare logical immediate	CLI	xD	3	T1 + 0.8
Set bits on masked	SBN	yA	3	T1 + 1.2
Set bits off masked	SBF	yB	3	T1 + 1.2
Test bits on masked	TBN	y8	3	T1 + 0.8
Test bits off masked	TBF	y9	3	T1 + 0.8
Store register	ST	y4	3	T1 + 1.4
Load register	L	y5	3	T1 + 1.2
Add to register	A	y6	3	T1 + 1.6
Branch on condition	BC	z0	3	1.6 (not taken), T1 (taken)
Load address	LA	z2	3	T1
Supervisor call	SVC	F4	5	1.8 + service time
Transfer	XFER	F5	5	1.8 + service time
Load program mode register	LPMR	F6	5	2.4
Jump on condition	JC	F2	6	1.6 (not taken), 2.0 (taken)

Hex Code Explanation

	Operand 1	Operand 2
x = 0	Direct	Direct
1	Direct	XR1
2	Direct	XR2
4	XR1	Direct
5	XR1	XR1
6	XR1	XR2
8	XR2	Direct
9	XR2	XR1
A	XR2	XR2
y = 3	Direct	
7	XR1	
B	XR2	
z = C		Direct
D		XR1
E		XR2

Operand Format Type Explanation

Type 1: Two-operand format: Length explicit or implied in both operands.
Type 2: Two-operand format: Length can be explicit in either operand but not in both. If length is not explicit in either operand, the assembler uses the implied length of operand 1.
Type 3: Two-operand format: Length cannot be specified.
Type 5: Two-operand format: Data operands are immediate data.
Type 6: Two-operand format: Operand 1 is used by the assembler to calculate a positive displacement; operand 2 is immediate data.

Time Explanation

L = Length.
T1 = 2.4 if direct, 2.0 if indexed.
T2 = 3.6 if direct, 2.8 if indexed, 3.2 if direct/indexed.
T3 = 1.2L1 if recomplemented and result does not equal minus zero, or 1.2 if recomplemented and result equals minus zero, else zero.
L2 = Length of operand 2.
L1 = Length of operand 1.

ASSEMBLER EXTENDED MNEMONIC OPERATION CODES

(2)

Instruction	Mnemonic Op Code	Hex Q-Code
Move hex character (MVX)		
Move to zone from zone	MZZ	X'00'
Move to numeric from zone	MNZ	X'02'
Move to zone from numeric	MZN	X'01'
Move to numeric from numeric	MNN	X'03'
Branch on condition (BC)		
Branch	B	X'87'
Branch high	BH	X'84'
Branch low	BL	X'82'
Branch equal	BE	X'81'
Branch not high	BNH	X'04'
Branch not low	BNL	X'02'
Branch not equal	BNE	X'01'
Branch overflow zoned	BOZ	X'88'
Branch overflow logical	BOL	X'A0'
Branch no overflow zoned	BNOZ	X'08'
Branch no overflow logical	BNOL	X'20'
Branch true	BT	X'10'
Branch false	BF	X'90'
Branch plus	BP	X'84'
Branch minus	BM	X'82'
Branch zero	BZ	X'81'
Branch not plus	BNP	X'04'
Branch not minus	BNM	X'02'
Branch not zero	BNZ	X'01'
Jump on condition (JC)		
Jump	J	X'87'
Jump high	JH	X'84'
Jump low	JL	X'82'
Jump equal	JE	X'81'
Jump not high	JNH	X'04'
Jump not low	JNL	X'02'
Jump not equal	JNE	X'01'
Jump overflow zoned	JOZ	X'88'
Jump overflow logical	JOL	X'A0'
Jump no overflow zoned	JNOZ	X'08'
Jump no overflow logical	JNOL	X'20'
Jump true	JT	X'10'
Jump false	JF	X'90'
Jump plus	JP	X'84'
Jump minus	JM	X'82'
Jump zero	JZ	X'81'
Jump not plus	JNP	X'04'
Jump not minus	JNM	X'02'
Jump not zero	JNZ	X'01'

OPERAND FORMATS

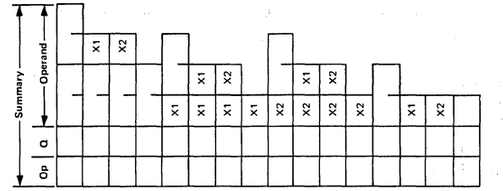
(3)

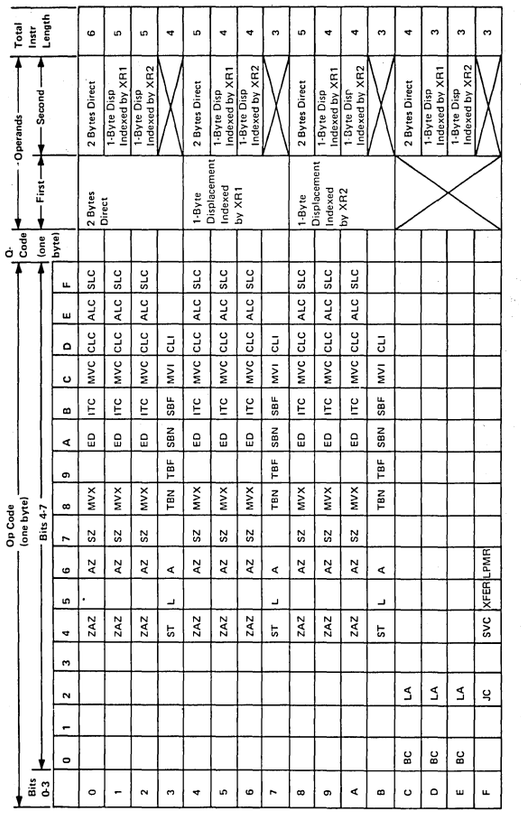
Type	Instructions	Possible Operand Format	
1	ZAZ, AZ, SZ	A,A	D(,R),A
		A,A(L)	D(,R),A(L)
		A,D(,R)	D(,R),D(,R)
		A,D(L,R)	D(,R),D(L,R)
		A(L),A	D(L,R),A
		A(L),A(L)	D(L,R),A(L)
		A(L),D(,R)	D(L,R),D(,R)
		A(L),D(L,R)	D(L,R),D(L,R)
2	MVC, CLC, ALC, SLC, ITC, ED	A,A	D(,R),A
		A(L),A	D(L,R),A
		A,A(L)	D(,R),A(L)
		A,D(,R)	D(,R),D(,R)
		A,D(L,R)	D(,R),D(L,R)
		A(L),D(,R)	D(L,R),D(,R)
	MVX	A,A(I)	D(,R),A(I)
		A,D(I,R)	D(,R),D(I,R)
		A(I),A	D(I,R),A
		A(I),D(,R)	D(I,R),D(,R)
3	MVI, CLI, SBN, SBF, TBN, TBF, BC	A,I	D(,R),I
	L, ST, A, LA	A,R	D(,R),R
5	SVC, XFER, LPMR	I,I	
6	JC	A,I	

Code	Meaning	Acceptable Form
A	Address	Relocatable expression, absolute expression, or self-defining value
D	Displacement	Relocatable expression, absolute expression, or self-defining value
L	Length	Absolute expression or self-defining value
R	Register	Absolute expression or self-defining value
I	Immediate data (bit masks, condition bit masks, or control bits to be used in the instruction)	Absolute expression or self-defining value

(4)

MAIN STORAGE INSTRUCTION FORMATS





PROGRAM STATUS REGISTER

Instruction	Condition	Binary Overflow (Bit 2)	Test False (Bit 3)	Decimal Overflow (Bit 4)	High (Bit 5)	Low (Bit 6)	Equal (Bit 7)
Zero and add zoned decimal	Set				Operand 2 result positive	Operand 2 result negative	Operand 2 result 0
	Reset				Operand not positive	Operand not negative	Operand not 0
Add and subtract zoned decimal	Set			Result overflows	Result positive	Result negative	Operand 2 result 0
	Reset				Result negative or 0	Result positive or 0	Result not 0
Edit[1]	Set				Operand 2 positive	Operand 2 negative	Operand 2 is 0
	Reset				Operand 2 not positive	Operand 2 not negative	Operand 2 not 0
Compare logical characters	Set				Operand 1 is > operand 2	Operand 1 is < operand 2	Operand 1 = operand 2
	Reset				Operand 1 is not > operand 2	Operand 1 is not < operand 2	Operands are not equal
Compare logical immediate	Set				Operand 1 is > immediate data	Operand 1 is < immediate data	Operand 1 is = immediate data
	Reset				Operand 1 is not > immediate data	Operand 1 is not < immediate data	Operand 1 is not = immediate data

[1] See *IBM System/34 Functions Reference*, SA21-9243, for more information about the Edit instruction.

PROGRAM STATUS REGISTER (continued)

Instruction	Condition	Binary Overflow (Bit 2)	Test False (Bit 3)	Decimal Overflow (Bit 4)	High (Bit 5)	Low (Bit 6)	Equal (Bit 7)
Add logical characters	Set	Carry out			Carry out and result not 0	No carry and result not 0	Result 0
	Reset	Reset at end of instruction execution			No carry or result 0	Carry out or result 0	Result not 0
Subtract logical characters	Set				Operand 1 is > operand 2	Operand 1 is < operand 2	Result 0
	Reset				Operand 1 is not > operand 2	Operand 1 is not < operand 2	Result not 0
Add to register	Set	Carry out			Carry out and result not 0	No carry and result not 0	Result 0
	Reset	Reset at end of instruction execution			No carry or result 0	Carry out or result 0	Result not 0
Test bits on			Tested bits are not all 1's				
Test bits off			Tested bits are not all 0's				

PROGRAM STATUS REGISTER (continued)

Instruction	Condition	Binary Overflow (Bit 2)	Test False (Bit 3)	Decimal Overflow (Bit 4)	High (Bit 5)	Low (Bit 6)	Equal (Bit 7)
Branch or jump on condition	Set						
	Reset		Reset if tested	Reset if tested			
Load register (PSR)	Set	Set if loaded bit 10 is on	Set if loaded bit 11 is on	Set if loaded bit 12 is on	Set if loaded bits 14 and 15 are off	Set if loaded bit 15 is off and bit 14 is on	Set if loaded bit 15 is on
	Reset	Reset if loaded bit 10 is off	Reset if loaded bit 11 is off	Reset if loaded bit 12 is off	Reset if bit 15 is on or if bit 15 is off and bit 14 is on	Reset if loaded bit 15 is on or if bits 14 and 15 are off	Reset if loaded bit 15 is off
System reset	Set						Set on if equal
	Reset	Binary overflow reset	Test false reset	Decimal overflow reset	High reset	Low reset	

HEXADECIMAL AND DECIMAL CONVERSION

To find the decimal value of a hexadecimal number, locate the hex number and its decimal equivalent for each position. Add these values to obtain the decimal number. To find the hex value of a decimal number, locate the next lower decimal number and its hex equivalent. Use the decimal difference to obtain the remaining hex numbers until the entire number is developed.

Conversion Table

Byte				Byte				Byte			
0123		4567		0123		4567		0123		4567	
Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec
0	0	0	0	0	0	0	0	0	0	0	0
1	1,048,536	1	65,536	1	4,096	1	256	1	16	1	1
2	2,097,152	2	131,072	2	8,192	2	512	2	32	2	2
3	3,145,728	3	196,608	3	12,288	3	768	3	48	3	3
4	4,194,304	4	262,144	4	16,384	4	1,024	4	64	4	4
5	5,242,880	5	327,680	5	20,480	5	1,280	5	80	5	5
6	6,291,456	6	393,216	6	24,576	6	1,536	6	96	6	6
7	7,340,032	7	458,752	7	28,672	7	1,792	7	112	7	7
8	8,388,608	8	524,288	8	32,768	8	2,048	8	128	8	8
9	9,437,184	9	589,824	9	36,864	9	2,304	9	144	9	9
A	10,485,760	A	655,360	A	40,960	A	2,560	A	160	A	10
B	11,534,336	B	720,896	B	45,056	B	2,816	B	176	B	11
C	12,582,912	C	788,432	C	49,152	C	3,072	C	192	C	12
D	13,631,488	D	851,968	D	53,248	D	3,328	D	208	D	13
E	14,680,064	E	917,504	E	57,344	E	3,584	E	224	E	14
F	15,728,640	F	983,040	F	61,440	F	3,840	F	240	F	15
6		5		4		3		2		1	

TASK ATR VALUE EQUATED TO REAL MAIN STORAGE ADDRESS

ATR Value	Real Main Storage Address	ATR Value	Real Main Storage Address	ATR Value	Real Main Storage Address	ATR Value	Real Main Storage Address
00	000000	10	008000	20	010000	30	018000
01	000800	11	008800	21	010800	31	018800
02	001000	12	009000	22	011000	32	019000
03	001800	13	009800	23	011800	33	019800
04	002000	14	00A000	24	012000	34	01A000
05	002800	15	00A800	25	012800	35	01A800
06	003000	16	00B000	26	013000	36	01B000
07	003800	17	00B800	27	013800	37	01B800
08	004000	18	00C000	28	014000	38	01C000
09	004800	19	00C800	29	014800	39	01C800
0A	005000	1A	00D000	2A	015000	3A	01D000
0B	005800	1B	00D800	2B	015800	3B	01D800
0C	006000	1C	00E000	2C	016000	3C	01E000
0D	006800	1D	00E800	2D	016800	3D	01E800
0E	007000	1E	00F000	2E	017000	3E	01F000
0F	007800	1F	00F800	2F	017800	3F	01F800

IBM®

International Business Machines Corporation

General Systems Division
4111 Northside Parkway N.W.
P.O. Box 2150
Atlanta, Georgia 30301
(U.S.A. only)

General Business Group/International
44 South Broadway
White Plains, New York 10601
U.S.A.
(International)

Printed in USA

GX21-7674-1

MESSAGE ISSUING MODULE ID

Component ID	Functional Description
AS	Assembler
AU	Auto report
BA	$BACK utility
BI	$BICR utility
BM	$BMENU utility
BS	BSC (binary synchronous communication)
BU	$BUILD utility
CA	Scheduler allocation
CB	COBOL
CC	Command processor commands
CF	$SETCF utility
CI	Scheduler initiation
CL	Scheduler logical I/O
CM	Command processor console management interface
CN	$CNFIG utility
CO	$COPY utility
CP	Command processor general functions
CS	Scheduler services
CT	Scheduler termination
DC	$DCSUP and $DCFUP utilities
DD	Disk data management
DE	$DELET utility
DF	DFU (data file utility)
DM	Common data management
DP	Printer data management
DR	Diskette data management
DU	$DUPRD utility
DW	Work station data management interface
ER	$ERAP utility
FB	$FBLD utility
FE	CE utility service aids
FO	FORTRAN
FR	$FREE utility
FS	FORTRAN scientific
GS	Sort
HF	History file access
HI	$HIST utility
IN	$INIT utility
IP	Printer IOS intercept
LA	$LABEL utility
LN	Linkage editor
LO	$LOAD utility
MA	$MAINT utility
MC	Concurrent maintenance for main storage
MG	$MGBLD utility
MI	MICR data management
MP	Macro processor
MR	MRJE (MULTI-LEAVING remote job entry)
MS	Main store IPL
NU	Nucleus functions: NUBE Serial printer error recovery NUCM Concurrent maintenance NUER Machine check error, nucleus error NUFD Disk error recovery NUI0 Diskette error recovery NUMR MICR (magnetic ink character recognition) NUTE Line printer error recovery NUWE Work station error recovery
OL	Overlay linkage editor
PA	$PACK utility
PG	RPG II (subroutines)
PR	Security functions
PT	#PTFLOG (PTF log module)
QJ	Queued job stream
RE	$RENAM utility
RP	RPG II
SE	SEU (source entry utility)
SF	$SFGR utility
SP	Spool
ST	System test function
SV	Main store supervisor
US	Syntax checker
WD	Work station data management
WS	WSU (work station utility)

SVC	Label	Byte (hex) R	Byte (hex) Q	Inline Parameters Bytes (0-3)	Inline Parameters Description
General wait	SVCGWAIT	00	02	0-1	General wait mask
General post	SVCGPOST	01	00	0-1	General wait mask post code
Wait	SVCWAIT	02	04 or 02		Not used
Post	SVCPOST	03	02	0	Queue header where posted action control element can be found (left byte)
				1	Bit: X'80'=1 — Do not preempt task issuing event post SVC X'40'=1 — Priority queue action control element to complete queue of task being posted Bits 4-7: Completion code (0-F)
Transfer control/system transient	SVCXFER	04	02	0	Offset in transient/transfer control table
Free current request block	SVCUNSTK	05	00		Not used
Assign system queue space	SVCASSGN	06	02 or 01	0-1 2	Length of area to assign Type of request: X'80'=1 — Queue area to task control block X'01'=0 — Use work station queue space X'01'=1 — Use system queue space
Free assigned areas	SVCFREE	07	00	0-1 2	Length of area to free Type of request: X'80'=1 — Area has been queued by assign X'01'=0 — Use work station queue space X'01'=1 — Use system queue space
Increase system event counter	SVCISEC	08	00	0	Offset of system event counter to be increased
Sense Address/Data switches	SVCSNSSW	09	00		Not used
Assign	SVCASGN	0A	02 or 01	0	Type of request: X'80'=1 — Queue area to task control block X'01'=0 — Use work station queue space X'01'=1 — Use system queue space

SVC	Label	Byte (hex) R	Byte (hex) Q	Inline Parameters Bytes (0-3)	Inline Parameters Description
Post action controller status word	SVCPPSVC	0B	00	0	Mask value associated with action controller routine to be posted
Load ATR	SVCLDATR	0C	00		Not used
Set program mode register	SVCPMR	0D	00	0	X'80' = Disable task dispatching X'70' = Reserved X'08' = Turn on instruction address register translation (see note) X'04' = Turn on operand 2 address translation X'02' = Turn on operand 1 address translation X'01' = Change from privileged to nonprivileged mode
				1	X'80' = Enable task dispatching X'70' = Reserved X'08' = Turn off instruction address register (see note) X'04' = Turn off operand 2 address translation X'02' = Turn off operand 1 address translation *Note:* Altering the instruction address register translation results in branching from the current translation to the same address in the opposite translation.
Queue/dequeue	SVCQUEUE	0E	00	0	Queue header displacement from start of system queue headers (for system request) (left byte)
				1	Displacement into control block of queuing field (0-255) (left byte)
				2	Queuing indicators and priority field displacement: X'80' = 1—Priority request X'40' = 1—System request (system queue header passed) X'20' = 1—Dequeue request X'10' = 1—Last-in, first-out request Bits 4-7: Priority field displacement (0-15)

SVC	Label	Byte (hex) R	Byte (hex) Q	Inline Parameters Bytes (0-3)	Inline Parameters Description
System control block access	SVCSCB	0F	00	0	Area and function
				1	Field displacement in area X'30' = Direct area 0-3 00—Direct area 0 01—Direct area 1 10—Direct area 2 11—Direct area 3 X'02' = 1—Queue header request X'01' = 1—Put request
Main storage transient scheduler	SVCXIENT	10	00	0-1	Address in main storage of a 4-byte (SSN) transient transfer control table entry (may be in either real or translated storage; however, it must be the same translation as the caller's instruction address register).
Main storage transient exit	SVCEXIT	11	00		Not used
Get page	SVCGETP	12	02	0-1	Address of where to store last logical address plus 1 of storage allocated
Free page	SVCFREEP	13	00		Not used
Interval timer (usable from control storage only)	SVCTIN SVCTID SVCTIR	14 15 16	00 00 00		Not used *Note:* R-byte =: 14 — Enqueue 15 — Dequeue 16 — Remainder
Asynchronous task wait	SVCTKWT	17	00	0	Wait mask
Set transient area to not busy	SVCXNTOF	18	00		Not used
Post action control element	SVCPOSTA	19	00	0	Queue header displacement where action control element can be found (left byte)
				1	Completion code: X'80'=1 — Do not preempt Bits 4-7: Completion code X'40'=1 — Queue last-in, first-out to task control block complete queue
Log trace information	SVCLOG	1A	00		Not used
Scan system queue	SVCQSCAN	1B	00	0	Displacement within control block where argument is located (left byte)
				1	Chain field displacement (left byte)

SVC	Label	Byte (hex) R	Byte (hex) Q	Inline Parameters Bytes (0-3)	Inline Parameters Description
Task post	SVCTPOST	1D	00	0	Task post condition
Task wait	SVCTWAIT	1E	02	0	Wait condition to be set on in task control block field TCBSTAT2
Interval timer interrupt handler (usable from control storage only)	SVCTIH	1F			Not used
Alter quiesce counter	SVCQS	20		0 1	Not used X'01' = Decrease quiesce counter
Resource enqueue/dequeue	SVCRENQ or SVCRDEQ	21	02 or 01	0	Share level: X'80'=1 — Enqueue request X'40'=1 — NEP request Bits 6-7: 00=Shares with 0, 1, and 2 01=Shares with 0 and 2 10=Shares with 0 11=Does not share
Dump main storage/terminate task	SVCDUMP	22	00	0-1	Abend MIC
Test and set	SVCTEST	23	04 02 01	0	Bit value to be tested
Task control block priority queue	SVCPRIQ	24	00	0	New priority
Asynchronous task ready check	SVCRDYCK	25	00		Not used
Prepare print buffer	SVCPREP	26			Not used
Dispatch	SVCDSPCH	27			Not used
	Reserved	28			
Sector enqueue/dequeue	SVCSQ	29	04		Not used
Quiesce counter wait	SVCQWAIT	2C	00		Not used
Delayed SVCs:					
Disk IOS	SVCFD	40	10 08 04 02 01		Not used

(13)

SVC	Label	Byte (hex) R	Byte (hex) Q	Inline Parameters Bytes (0-3)	Inline Parameters Description
Diskette IOS	SVCIO	41	10 08 04 02 01		Not used
Work station printer/line printer IOCH	SVCPT	42	10 08 04 02 01		Not used
Work station terminal IOCH	SVCWSC	43	10 08 04 02 01		Not used
Data communications IOCH	SVCCOMM	44	10 08 04 02 01		Not used
I/O transient request	SVCIOXNT	45	10 08 04 02 01		Not used
	Reserved	46-4B			
Action control element build and queue	SVCPQSVC	4C	10 08 04 01	0	Queue header displacement (left byte) X'01' = Return action control element address
	Reserved	4D-4F			
Control storage scheduler	SVCCXNT	50	02	0 1-2	Control storage transient ID Input to the transient
Task work area access	SVCTWA	51	02	0 1 2	Type: X'40' = 1—Real data address X'04' = 0—Task work area request X'04' = 1—Work station area request X'02' = 1—System request X'01' = 0—Get request X'01' = 1—Put request Key (0-59) Number of sectors

(14)

SVC	Label	Byte (hex) R	Byte (hex) Q	Inline Parameters Bytes (0-3)	Inline Parameters Description
Main storage relocation loader	SVCLOAD	52	04 or 02	0	Value determines type of request: X'01' = Load by relative address—Adds the task loader disk address to the relative address passed in the user's parameter list. The resulting address is the location of the desired module on the disk. The module is loaded at its link-edit address. Control is returned to the calling program. X'02' = Load to address—Reads the module in storage and returns control to the calling program. X'04' = Fetch—Adds the task relocation factor to the module's link-edit address and, using the resulting value as the load address, reads the module into storage and passes control to the module's start control address. X'06' = Fetch to address—Reads the module into storage and passes control to the module's start control address. X'0A' = System load to address—Updates the task relocation factor and disk address values in the task's task control block from the loader's parameter list. Reads the module into storage and returns control to the calling program. X'0E' = System fetch to address—Updates the task relocation factor and disk address values in the task's task control block from the loader's parameter list. Reads the module into storage and passes control to its start control address.

(15)

DISPLAY SCREEN CHARACTER SET

(16)

First Hex Character

Second Hex Character	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0					SP	&	-	Ø	ø			⊙	{	}	\	⊙
1						é	/	É	a	j	~	1	A	J		1
2					â	ê	Â	Ê	b	k	s	2	B	K	S	2
3					ä	ë	Ä	Ë	c	l	t	3	C	L	T	3
4					à	è	À	È	d	m	u	4	D	M	U	4
5					á	í	Á	Í	e	n	v	5	E	N	V	5
6		←			ã	î	Ã	Î	f	o	w	6	F	O	W	6
7					å	ï	Å	Ï	g	p	x	7	G	P	X	7
8					ç	ì	Ç	Ì	h	q	y	8	H	Q	Y	8
9					ñ	ß	Ñ	`	i	r	z	9	I	R	Z	9
A					¢	!	¦	:								
B					.	$	,	#					ô	û	Ô	Û
C		¯			<	*	%	@		æ			ö	ü	Ö	Ü
D					(	)	–	'					ò	ù	Ò	Ù
E					+	;	>	=		Æ			ó	ú	Ó	Ú
F					|	¬	?	"		■			õ	ÿ	Õ	

Note: Display codes hex B0 through B9 designate negative numbers.

Second Edition (July 1978)

This is a major revision of, and obsoletes, GX21-7674-0. Because the changes and additions are extensive, this publication should be reviewed in its entirety. Changes are periodically made to the information herein; before using this publication in connection with the operation of IBM systems, refer to the latest *IBM System/34 Bibliography*, GH30-0231, for the editions that are applicable and current.

The purpose of this publication is to summarize frequently used reference material for System/34 programmers and engineers.

Publications are not stocked at the address below. Requests for copies of IBM publications and for technical information about the system should be made to your IBM representative or to the branch office serving your locality.

This publication could contain technical inaccuracies or typographical errors. Address your comments about this publication to IBM Corporation, Publications, Department 245, Rochester, Minnesota 55901. Comments become the property of IBM.