SPERRY UNIVAC COMPUTER SYSTEMS

SERIES 70
ASSEMBLY
REFERENCE CARD

INSTRUCTION SET

Mnemonic	Symbolic Operand Format	Machine Code	Type	Instruction Name
A	R1,D2(X2,B2)	5A	RX	Add
AD	R1,D2(X2,B2)	6A	RX	Add Normalized, Long
ADR	R1,R2	2A	RR	Add Normalized, Long (Register)
AE	R1,D2(X2,B2)	7A	RX	Add Normalized, Short
AER	R1,R2	3A	RR	Add Normalized, Short (Register)
AH	R1,D2(X2,B2)	4A	RX	Add Half Word
AL	R1,D2(X2,B2)	5E	RX	Add Logical
ALR	R1,R2	1E	RR	Add Logical (Register)
AP	D1(L1,B1),D2(L2,B2)	FA	SS	Add Decimal
AR	R1,R2	1A	RR	Add (Register)
AU	R1,D2(X2,B2)	7E	RX	Add Unnormalized, Short
AUR	R1,R2	3E	RR	Add Unnormalized, Short (Register)
AW	R1,D2(X2,B2)	6E	RX	Add Unnormalized, Long
AWR	R1,R2	2E	RR	Add Unnormalized, Long (Register)
BAL	R1,D2(X2,B2)	45	RX	Branch and Link
BALR	R1,R2	05	RR	Branch and Link Register
BC	M1,D2(X2,B2)	47	RX	Branch on Condition
BCR	M1,R2	07	RR	Branch on Condition (Register)
BCT	R1,D2(X2,B2)	46	RX	Branch on Count
BCTR	R1,R2	06	RR	Branch on Count (Register)
BXH	R1,R3,D2(B2)	86	RS	Branch on Index High
BXLE	R1,R3,D2(B2)	87	RS	Branch on Index Low or Equal
C	R1,D2(X2,B2)	59	RX	Compare (Word)
CD	R1,D2(X2,B2)	69	RX	Compare, Long
CDR	R1,R2	29	RR	Compare, Long (Register)
CE	R1,D2(X2,B2)	79	RX	Compare, Short
CER	R1,R2	39	RR	Compare, Short (Register)
CH	R1,D2(X2,B2)	49	RX	Compare Half Word
CKC	D1(B1)	9F	SI	Check Channel*
CL	R1,D2(X2,B2)	55	RX	Compare Logical (Word)
CLC	D1(L,B1),D2(B2)	D5	SS	Compare Logical (Character)
CLI	D1(B1),I2	95	SI	Compare Logical Immediate
CLR	R1,R2	15	RR	Compare Logical (Register)
CP	D1(L1,B1),D2(L2,B2)	F9	SS	Compare Decimal
CR	R1,R2	19	RR	Compare (Register)
CVB	R1,D2(X2,B2)	4F	RX	Convert to Binary
CVD	R1,D2(X2,B2)	4E	RX	Convert to Decimal
D	R1,D2(X2,B2)	5D	RX	Divide (Word)
DD	R1,D2(X2,B2)	6D	RX	Divide, Long
DDR	R1,R2	2D	RR	Divide, Long (Register)
DE	R1,D2(X2,B2)	7D	RX	Divide, Short
DER	R1,R2	3D	RR	Divide, Short (Register)
DIG	D1(B1),I2	83	SI	Diagnose
DP	D1(L1,B1),D2(L2,B2)	FD	SS	Divide Decimal
DR	R1,R2	1D	RR	Divide (Register)
ED	D1(L,B1),D2(B2)	DE	SS	Edit
EDMK	D1(L,B1),D2(B2)	DF	SS	Edit and Mark
EX	R1,D2(X2,B2)	44	RX	Execute
HDR	R1,R2	24	RR	Halve, Long (Register)
HDV	D1(B1)	9E	SI	Halt Device*
HER	R1,R2	34	RR	Halve, Short (Register)
IC	R1,D2(X2,B2)	43	RX	Insert Character
IDL	I2	80	SI	Idle*
ISK	R1,R2	08	RR	Insert Storage Key*
L	R1,D2(X2,B2)	58	RX	Load (Word)
LA	R1,D2(X2,B2)	41	RX	Load Address
LCDR	R1,R2	23	RR	Load Complement, Long (Register)
LCER	R1,R2	33	RR	Load Complement, Short (Register)
LCR	R1,R2	13	RR	Load Complement
LD	R1,D2(X2,B2)	68	RX	Load, Long

*Privileged Operation

EA-033-0-00 (APRIL 1974)

INSTRUCTION SET (cont)

Mnemonic	Symbolic Operand Format	Machine Code	Type	Instruction Name
LDR	R1,R2	28	RR	Load, Long (Register)
LE	R1,D2(X2,B2)	78	RX	Load, Short
LER	R1,R2	38	RR	Load, Short (Register)
LH	R1,D2(X2,B2)	48	RX	Load Half Word
LM	R1,R3,D2(B2)	98	RS	Load Multiple
LNDR	R1,R2	21	RR	Load Negative, Long (Register)
LNER	R1,R2	31	RR	Load Negative, Short (Register)
LNR	R1,R2	11	RR	Load Negative
LPDR	R1,R2	20	RR	Load Positive, Long (Register)
LPER	R1,R2	30	RR	Load Positive, Short (Register)
LPR	R1,R2	10	RR	Load Positive Register
LR	R1,R2	18	RR	Load Register
LSP	D1(L,B1),D2(B2)	D8	SS	Load Scratch Pad*
LTDR	R1,R2	22	RR	Load and Test, Long (Register)
LTER	R1,R2	32	RR	Load and Test, Short (Register)
LTR	R1,R2	12	–	Load and Test
M	R1,D2(X2,B2)	5C	RX	Multiply (Word)
MD	R1,D2(X2,B2)	6C	RX	Multiply, Long
MDR	R1,R2	2C	RR	Multiply, Long (Register)
ME	R1,D2(X2,B2)	7C	RX	Multiply, Short
MER	R1,R2	3C	RR	Multiply, Short (Register)
MH	R1,D2(X2,B2)	4C	RX	Multiply Half Word
MP	D1(L1,B1),D2(L2,B2)	FC	SS	Multiply Decimal
MR	R1,R2	1C	RR	Multiply (Register)
MVC	D1(L,B1),D2(B2)	D2	SS	Move Character
MVI	D1(B1),I2	92	SI	Move Immediate
MVN	D1(L,B1),D2(B2)	D1	SS	Move Numerics
MVO	D1(L1,B1),D2(L2,B2)	F1	SS	Move with Offset
MVZ	D1(L,B1),D(B2)	D3	SS	Move Zones
N	R1,D2(X2,B2)	54	RX	And Logical (Word)
NC	D1(L,B1),D2(B2)	D4	SS	And Logical (Characters)
NI	D1(B1),I2	94	SI	And Logical Immediate
NR	R1,R2	14	RR	And Logical (Register)
O	R1,D2(X2,B2)	56	RX	Or Logical (Word)
OC	D1(L,B1),D2(B2)	D6	SS	Or Logical (Character)
OI	D1(B1),I2	96	SI	Or Logical Immediate
OR	R1,R2	16	RR	Or Logical (Register)
PACK	D1(L1,B1),D2(L2,B2)	F2	SS	Pack
PC	D1(B1),I2	82	SI	Program Control*
RDD	D1(B1),I2	85	SI	Read Direct*
S	R1,D2(X2,B2)	5B	RX	Subtract (Word)
SD	R1,D2(X2,B2)	6B	RX	Subtract Normalized, Long
SDR	R1,R2	2B	RR	Subtract Normalized, Long (Register)
SDV	D1(B1)	9C	SI	Start Device*
SE	R1,D2(X2,B2)	7B	RX	Subtract Normalized, Short
SER	R1,R2	3B	RR	Subtract Normalized, Short (Register)
SH	R1,D2(X2,B2)	4B	RX	Subtract Half Word
SL	R1,D2(X2,B2)	5F	RX	Subtract Logical
SLA	R1,D2(B2)	8B	RS	Shift Left Single
SLDA	R1,D2(B2)	8F	RS	Shift Left Double
SLDL	R1,D2(B2)	8D	RS	Shift Left Double Logical
SLL	R1,D2(B2)	89	RS	Shift Left Single Logical
SLR	R1,R2	1F	RR	Subtract Logical (Register)
SP	D1(L1,B1),D2(L2,B2)	FB	SS	Subtract Decimal
SPM	R1,R2	04	RR	Set Program Mask
SR	R1,R2	1B	RR	Subtract (Register)
SRA	R1,D2(B2)	8A	RS	Shift Right Single
SRDA	R1,D2(B2)	8E	RS	Shift Right Double
SRDL	R1,D2(B2)	8C	RS	Shift Right Double Logical
SRL	R1,D2(B2)	88	RS	Shift Right Single Logical

*Privileged Operation

INSTRUCTION SET (cont)

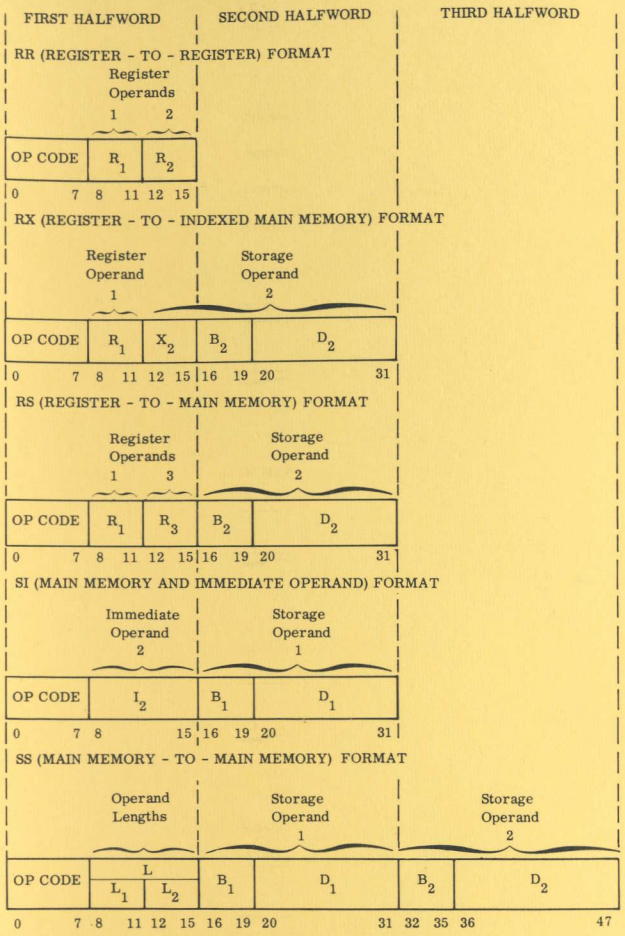
Mnemonic	Symbolic Operand Format	Machine Code	Type	Instruction Name
SSK	R1,R2	08	RR	Set Storage Key*
SSP	D1(L,B1),D2(B2)	D0	SS	Store Scratch Pad*
ST	R1,D2(X2,B2)	50	RX	Store (Word)
STC	R1,D2(X2,B2)	42	RX	Store (Character)
STD	R1,D2(X2,B2)	60	RX	Store Long
STE	R1,D2(X2,B2)	70	RX	Store Short
STH	R1,D2(X2,B2)	40	RX	Store Half Word
STM	R1,R3,D2(B2)	90	RS	Store Multiple
SU	R1,D2(X2,B2)	7F	RX	Subtract Unnormalized, Short
SUR	R1,R2	3F	RR	Subtract Unnormalized, Short (Register)
SVC	I	0A	RR	Supervisor Call
SW	R1,D2(X2,B2)	6F	RX	Subtract Unnormalized, Long
SWR	R1,R2	2F	RR	Subtract Unnormalized, Long (Register)
TDV	D1(B1)	9D	SI	Test Device*
TM	D1(B1),I2	91	SI	Test Under Mask
TR	D1(L,B1),D2(B2)	DC	SS	Translate
TRT	D1(L,B1),D2(L2,B2)	DD	SS	Translate and Test
TS	D1(B1),I2	93	–	Test and Set
UNPK	D1(L1,B1),D2(L2,B2)	F3	SS	Unpack
WRD	D1(B1),I2	84	SI	Write Direct*
X	R1,D2(X2,B2)	57	RX	Exclusive Or (Word)
XC	D1(L,B1),D2(B2)	D7	SS	Exclusive Or (Character)
XI	D1(B1),I2	97	SI	Exclusive Or, Immediate
XR	R1,R2	17	RR	Exclusive Or (Register)
ZAP	D1(L1,B1),D2(L2,B2)	F8	SS	Zero and Add Decimal

*Privileged Operation

MNEMONIC OP CODES BY FORMAT

Instruction	Machine Format
AP, CP, DP, MP, MVO, PACK, SP, UNPK, ZAP.	SS FORMAT (1)
CLC, ED, EDMK, LSP, MVC, MVN, MVZ, NC, OC, SSP, TR, TRT, XC.	SS FORMAT (2)
CKC, CLI, DIG, HDV, IDL, MVI, NI, OI, PC, RDD, SDV, TDV, TM, WRD, XI.	SI FORMAT
LM, SLA, SLDA, SLDL, SLL, S SRA, SRDA, SRDL, SRL, STM, BXH, BXLE.	RS FORMAT
A, AD, AE, AH, AL, AU, AW, BAL, BC, BCT, C, CD, CE, CH, CL, CVB, CVD, D, DD, DE, EX, IC, L, LA, LD, LE, LH, M, MD, ME, MH, N, O, S, SD, SE, SH, SL, ST, STC, STD, STE, STH, SU, SW, X.	RX FORMAT
ADR, AER, ALR, AR, AUR, AWR, BALR, BCR, BCTR, CDR, CER, CLR, CR, DDR, DER, DR, HDR, HER, ISK, LCDR, LCER, LCR, LDR, LER, LNDR, LNER, LPDR, LPER, LPR, LR, LTDR, LTER, LTR, MDR, MER, MR, NR, OR, SDR, SER, SLR, SPM, SR, SSK, SUR, SVC, SWR, SR, LNR.	RR FORMAT

MACHINE INSTRUCTION FORMATS



LEGEND:

B_1 - Four-bit general register containing base address of first operand.

B_2 - Four-bit general register containing base address of second operand.

D_1 - Twelve-bit field designating address of leftmost byte of first operand.

D_2 - Twelve-bit field designating address of leftmost byte of second operand. Eight-bit field designating

I_2 - Immediate operand.

L - Eight-bit operand length specification designating the number of additional bytes that are to the right of the first and the second operand address.

L_1 - Four-bit operand length specification designating one less than length of first operand.

L_2 - Four-bit operand length specification designating one less than length of second operand.

R_1 - Four-bit general register containing operand 1.

R_2 - Four-bit general register containing operand 2.

R_3 - Four-bit general register containing operand 3.

X_2 - Four-bit general register used for indexing.

ONDITION CODE SETTINGS

Instruction Set	Condition Code			
	8	4	2	1
1. After Add/Subtract Logical (AL, ALR, SL, SLR).	Result = 0 No carry	Result ≠ 0 No carry	Result = 0 Carry	Result ≠ 0 Carry
2. After Fixed Point, Floating Point, Decimal Add And Subtract (A, AD, ADR, AE, AER, AH, AP, AR, AU, AUR, AW, AWR, S, SD, SDR, SE, SER, SH, SP, SR, SU, SUR, SW, SWR, ZAP). Left Shift (SLA, SLDA); and LCR instructions.	Result = 0	Result < 0	Result > 0	Overflow
3. After Compare A:B instructions (C, CD, CDR, CE, CER, CH, CL, CLC, CLI, CLR, CP, CR).	A = B	A < B	A > B	
4. After Edit (ED, EDMK); Right Shift (SRA, SRDA); and Load and Test (LCDR, LCER, LTDR, LDER, LTR) instructions.	Result = 0	Result < 0	Result >0	
5. After Load Positive (LPDR, LPER, LPR) instructions.	Result = 0		Result > 0	Overflow (LPR only)
6. After Load Negative (LNDR, LNER, LNR) instructions.	Result = 0	Result < 0		
7. After Logical And (N, NC, NI, NR); Logical OR (O, OC, OI, OR); Exclusive OR (X, XC, XI, XR) instructions.	Result = 0	Result ≠ 0		
8. After TM instruction.	Selected bits = 0	Selected bits mixed 0's and 1's.		Selected bits all 1's.
9. After TRT instruction.	All function bytes = 0	Non-zero function byte.	Last function byte non-zero.	

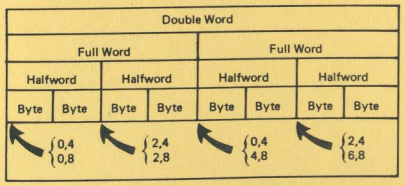
ONDITION CODE SUMMARY

Bit position	2^3	2^2	2^1	2^0
After test-under-mask	Zeros	Mixed		Ones
Others	=	<	>	Overflow

DITING MASK SYMBOLS

Name	Hexadecimal Code	Function
Digit Select	20	Specifies digit position.
Start Significance	21	Stops replacement of leading zeros, is also a digit select.
Field Separator	22	Indicates new field.
Filler	Any	Replaces leading zeros.
Insertion Character	Any	Inserted in result.

ONDITION NO OPERATION (CNOP) DOUBLE WORD ALIGNMENT



EXTENDED MNEMONIC CODES

Extended Code		Meaning	Machine-Instruction	
B	D2(X2,B2)	Branch Unconditional	BC	15,D2(X2,B2)
BR	R2	Branch Unconditional (RR format)	BCR	15,R2
NOP	D2(X2,B2)	No Operation	BC	0,D2(X2,B2)
NOPR	R2	No Operation (RR format)	BCR	0,R2
Used After Compare Instructions				
BH	D2(X2,B2)	Branch on High	BC	2,D2(X2,B2)
BL	D2(X2,B2)	Branch on Low	BC	4,D2(X2,B2)
BE	D2(X2,B2)	Branch on Equal	BC	8,D2(X2,B2)
BNH	D2(X2,B2)	Branch on Not High	BC	13,D2(X2,B2)
BNL	D2(X2,B2)	Branch on Not Low	BC	11,D2(X2,B2)
BNE	D2(X2,B2)	Branch on Not Equal	BC	7,D2(X2,B2)
*BRH	R2	Branch on High (RR format)	BCR	2,R2
*BRL	R2	Branch on Low (RR format)	BCR	4,R2
*BRE	R2	Branch on Equal (RR format)	BCR	8,R2
*BRNH	R2	Branch on Not High (RR format)	BCR	13,R2
*BRNL	R2	Branch on Not Low (RR format)	BCR	11,R2
*BRNE	R2	Branch on Not Equal (RR format)	BCR	7,R2
Used After Arithmetic Instructions				
BO	D2(X2,B2)	Branch on Overflow	BC	1,D2(X2,B2)
BP	D2(X2,B2)	Branch on Plus	BC	2,D2(X2,B2)
BM	D2(X2,B2)	Branch on Minus	BC	4,D2(X2,B2)
BZ	D2(X2,B2)	Branch on Zero	BC	8,D2(X2,B2)
*BNP	D2(X2,B2)	Branch on Not Plus	BC	13,D2(X2,B2)
*BNM	D2(X2,B2)	Branch on Not Minus	BC	11,D2(X2,B2)
*BNZ	D2(X2,B2)	Branch on Not Zero	BC	7,D2(X2,B2)
*BRO	R2	Branch on Overflow (RR format)	BCR	1,R2
*BRP	R2	Branch on Plus (RR format)	BCR	2,R2
*BRM	R2	Branch on Minus (RR format)	BCR	4,R2
*BRZ	R2	Branch on Zero (RR format)	BCR	8,R2
*BRNP	R2	Branch on Not Plus (RR format)	BCR	13,R2
*BRNM	R2	Branch on Not Minus (RR format)	BCR	11,R2
*BRNE	R2	Branch on Not Equal (RR format)	BCR	7,R2
Used After Test Under Mask Instructions				
BO	D2(X2,B2)	Branch if Ones	BC	1,D2(X2,B2)
BM	D2(X2,B2)	Branch if Mixed	BC	4,D2(X2,B2)
BZ	D2(X2,B2)	Branch if Zeros	BC	8,D2(X2,B2)
*BNO	D2(X2,B2)	Branch if Not Ones	BC	14,D2(X2,B2)
*BNM	D2(X2,B2)	Branch if Not Mixed	BC	11,D2(X2,B2)
*BNZ	D2(X2,B2)	Branch if Not Zero	BC	7,D2(X2,B2)

*VMS Assembler Only

POWERS OF 16 AND POWERS OF 2

POWERS OF 16 16^n	n	POWERS OF 2 2^n	n
1	0	512	9
16	1	1 024	10
256	2	2 048	11
4 096	3	4 096	12
65 536	4	8 192	13
1 048 576	5	16 384	14
16 777 216	6	32 768	15
268 435 456	7	65 536	16
4 294 967 296	8	131 072	17
68 719 476 736	9	262 144	18
1 099 511 627 776	10	524 288	19
17 592 186 044 416	11	1 048 576	20
281 474 976 710 656	12	2 097 152	21
4 503 599 627 370 496	13	4 194 304	22
72 057 594 037 927 936	14	8 388 608	23
1 152 921 504 606 846 976	15	16 777 216	24

INTERRUPT CONDITIONS

Priority	Interrupt Condition	Flag *Bit	State Initiated	Weight Hexadecimal
1	Power Failure	2^0	P_4	00
2	Machine Check	2^1	P_4	04
3	External Signal No. 1	2^2	P_3	08
4	External Signal No. 2	2^3	P_3	0C
5	External Signal No. 3	2^4	P_3	10
6	External Signal No. 4	2^5	P_3	14
7	External Signal No. 5	2^6	P_3	18
8	External Signal No. 6	2^7	P_3	1C
9	Interval Timer	2^8	P_3	20
10	Selector Channel No. 1	2^9	P_3	24
11	Selector Channel No. 2	2^{10}	P_3	28
12	Selector Channel No. 3	2^{11}	P_3	2C
13	Selector Channel No. 4	2^{12}	P_3	30
14	Selector Channel No. 5	2^{13}	P_3	34
15	Selector Channel No. 6	2^{14}	P_3	38
16	Multiplexor Channel	2^{15}	P_3	3C
17	Elapsed Time Clock	2^{16}	P_3	40
18	Console Interrupt Request	2^{17}	P_3	44
19	Paging Error	2^{18}	P_3	48
20	Paging Queue	2^{19}	P_3	4C
21	Supervisor Call Instruction	2^{20}	P_3	50
22	Privileged Operation	2^{21}	P_3	54
23	Op-Code Trap	2^{22}	P_3	58
24	Address Error (Protect, Addressing, Specification)	2^{23}	P_3	5C
25	Data Error	2^{24}	P_3	60
26	Exponent Overflow	2^{25}	P_3	64
27	Divide Error	2^{26}	P_3	68
28	Significant Error	2^{27}	P_3	6C
29	Exponent Underflow	2^{28}	P_3	70
30	Decimal Overflow	2^{29}	P_3	74
31	Fixed Point Overflow	2^{30}	P_3	78
32	Test Mode	2^{31}	P_3	7C

*2^0 = The rightmost bit in the Interrupt Flag register.

EXPLICIT AND IMPLIED OPERAND FORMATS

Type	Explicit Address	Implied Address
RX	D2(X2,B2) D2(0,B2)	S2(X2) S2
RS	D2 (B2)	S2
SI	D1 (B1)	S1
SS	D1 (L1,B1) D1 (L,B1) D1 (L2,B2)	S1 (L1) S1 (L) S2 (L2)

HEXADECIMAL ADDITION TABLE

BASE 10	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
10	16	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	1	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10
2	2	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11
3	3	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12
4	4	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13
5	5	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14
6	6	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15
7	7	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16
8	8	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17
9	9	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18
10	A	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19
11	B	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A
12	C	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B
13	D	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C
14	E	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D
15	F	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E

HEXADECIMAL TO DECIMAL CONVERSION TABLE

X	$X.16^5$	$X.16^4$	$X.16^3$	$X.16^2$	$X.16^1$	$X.16^0$	X
0	0	0	0	0	0	0	0
1	1,048,576	65,536	4,096	256	16	1	1
2	2,097,152	131,072	8,192	512	32	2	2
3	3,145,728	196,608	12,288	768	48	3	3
4	4,194,304	262,144	16,384	1,024	64	4	4
5	5,242,880	327,680	20,480	1,280	80	5	5
6	6,291,456	393,216	24,576	1,536	96	6	6
7	7,340,032	458,752	28,672	1,792	112	7	7
8	8,388,608	524,288	32,768	2,048	128	8	8
9	9,437,184	589,824	36,864	2,304	144	9	9
A	10,485,760	655,360	40,960	2,560	160	10	A
B	11,534,336	720,896	45,056	2,816	176	11	B
C	12,582,912	786,432	49,152	3,072	192	12	C
D	13,631,488	851,968	53,248	3,328	208	13	D
E	14,680,064	917,504	57,344	3,584	224	14	E
F	15,728,640	983,040	61,440	3,840	240	15	F
X	$X.16^5$	$X.16^4$	$X.16^3$	$X.16^2$	$X.16^1$	$X.16^0$	X

2 BYTE INSTRUCTIONS (RR)

Decimal	Hexadecimal	Mnemonic	EBCDIC	Instruction Action	EBCDIC Extended Hollerith EAM Card Code
0	00		0000 0000		12,0,9,8,1
1	01		0000 0001		12,9,1
2	02		0000 0010		12,9,2
3	03		0000 0011		12,9,3
4	04	SPM	0000 0100	$PSW_{34-39} \leftarrow c(R1)_{2-7}$	12,9,4
5	05	BALR	0000 0101	$R1 \leftarrow c(PSW)_{32-63}$; $\rightarrow c(R2)$	12,9,5
6	06	BCTR	0000 0110	$R1 \leftarrow c(R1)-1$; $\rightarrow c(R2)$ if $c(R1) \neq 0$	12,9,6
7	97	BCR	0000 0111	$\rightarrow c(R2)$ if (M1)=1	12,9,7
8	08	SSK*	0000 1000	Storage Key of $c(R2)_{8-20} \leftarrow c(R1)_{24-27}$	12,9,8
9	09	ISK*	0000 1001	$R1_{24-27} \leftarrow$ Storage Key of $c(R2)_{8-20}$	12,9,8,1
10	0A	SVC	0000 1010	Interrupt; $PSW(old)_{21-31} \leftarrow I$	12,9,8,2
11	0B		0000 1011		12,9,8,3
12	0C		0000 1100		12,9,8,4
13	0D		0000 1101		12,9,8,5
14	0E		0000 1110		12,9,8,6
15	0F		0000 1111		12,9,8,7
16	10	LPR	0001 0000	$R1 \leftarrow \lvert c(R2) \rvert$	12,11,9,8,1
17	11	LNR	0001 0001	$R1 \leftarrow -\lvert c(R2) \rvert$	11,9,1
18	12	LTR	0001 0010		11,9,2
19	13	LCR	0001 0011	$R1 \leftarrow -c(R2)$	11,9,3
20	14	NR	0001 0100	R1←c(R1) AND c(R2)	11,9,4
21	15	CLR	0001 0101		11,9,5
22	16	OR	0001 0110	R1←c(R1) OR c(R2)	11,9,6
23	17	XR	0001 0111	R1←c(R1) Ex. OR c(R2)	11,9,7
24	18	LR	0001 1000	R1←c(R2)	11,9,8
25	19	CR	0001 1001		11,9,8,1
26	1A	AR	0001 1010	R1←c(R1)+c(R2)	11,9,8,2
27	1B	SR	0001 1011	R1←c(R1)−c(R2)	11,9,8,3
28	1C	MR	0001 1100	[R1,R1+1]←c(R1+1)Xc(R2)	11,9,8,4
29	1D	DR	0001 1101	R1←Rem. of [c(R1),c(R1+1)] /c(R2) R1+1←Quot. of [c(R1),c(R1+1)] /c(R2)	11,9,8,5
30	1E	ALR	0001 1110	R1←c(R1)+c(R2)	11,9,8,6
31	1F	SLR	0001 1111	R1←c(R1)−c(R2)	11,9,8,7
32	20	LPDR	0010 0000	FPR1←$\lvert$c(FPR2)$\rvert$1	11,0,9,8,1
33	21	LNDR	0010 0001	FPR1←$-\lvert$c(FPR2)$\rvert$1	0,9,1
34	22	LTDR	0010 0010	FPR1←c(FPR2)	0,9,2
35	23	LCDR	0010 0011	FPR1←−c(FPR2)	0,9,3
36	24	HDR	0010 0100	FPR1←c(FPR2)/2	0,9,4
37	25		0010 0101		0,9,5
38	26		0010 0110		0,9,6
39	27		0010 0111		0,9,7
40	28	LDR	0010 1000	FPR1←c(FPR2)	0,9,8
41	29	CDR	0010 1001		0,9,8,1
42	2A	ADR	0010 1010	FPR1←c(FPR1)+c(FPR2)	0,9,8,2
43	2B	SDR	0010 1011	FPR1←c(FPR1)−c(FPR2)	0,9,8,3
44	2C	MDR	0010 1100	FPR1←c(FPR1)Xc(FPR2)	0,9,8,4
45	2D	DDR	0010 1101	FPR1←c(FPR1)/c(FPR2)	0,9,8,5
46	2E	AWR	0010 1110	FPR1←c(FPR1)+c(FPR2)	0,9,8,6
47	2F	SWR	0010 1111	FPR1←c(FPR1)−c(FPR2)	0,9,8,7
48	30	LPER	0011 0000	FPR1←$\lvert$c(FPR2)$\rvert$1	12,11,0,9,8,1
49	31	LNER	0011 0001	FPR1←$-\lvert$c(FPR2)$\rvert$1	9,1
50	32	LTER	0011 0010	FPR1←c(FPR2)	9,2
51	33	LCER	0011 0011	FPR1←−c(FPR2)	9,3
52	34	HER	0011 0100	FPR1←c(FPR2)/2	9,4
53	35		0011 0101		9,5
54	36		0011 0110		9,6
55	37		0011 0111		9,7
56	38	LER	0011 1000	FPR1←c(FPR2)	9,8
57	39	CER	0011 1001		9,8,1
58	3A	AER	0011 1010	FPR1←c(FPR1)+c(FPR2)	9,8,2
59	3B	SER	0011 1011	FPR1←c(FPR1)−c(FPR2)	9,8,3
60	3C	MER	0011 1100	FPR1←c(FPR1)Xc(FPR2)	9,8,4
61	3D	DER	0011 1101	FPR1←c(FPR1)/c(FPR2)	9,8,5
62	3E	AUR	0011 1110	FPR1←c(FPR1)+c(FPR2)	9,8,6
63	3F	SUR	0011 1111	FPR1←c(FPR1)−c(FPR2)	9,8,7

4 BYTE INSTRUCTIONS (RX)

Decimal	Hexadecimal	Mnemonic	EBCDIC	Instruction Action	EBCDIC Extended Hollerith EAM Card Code
64	40	STH	0100 0000		
65	41	LA	0100 0001	$R1_{8-31} \leftarrow S2$; $R1_{0-7} \leftarrow 0$	12,0,9,1
66	42	STC	0100 0010	$S2_{0-7} \leftarrow c(R1)_{24-31}$	12,0,9,2
67	43	IC	0100 0011	$R1_{24-31} \leftarrow c(S2)_{0-7}$	12,0,9,3
68	44	EX	0100 0100	Execute instr. c(S2), mod by $c(R1)_{24-31}$	12,0,9,4
69	45	BAL	0100 0101	$R1 \leftarrow c(PSW)_{32-63}$; $\leftarrow S2$	12,0,9,5
70	46	BCT	0100 0110	$R1 \leftarrow c(R1)-1$; $\leftarrow S2$ if $c(R1) \neq 0$	12,0,9,6
71	47	BC	0100 0111	$\leftarrow S2$ if (M1)=1	12,0,9,7
72	48	LH	0100 1000	$R1_{16-31} \leftarrow c(S2)_{0-15}$; $R1_{0-15} \leftarrow c(S2)_0$	12,0,9,8
73	49	CH	0100 1001		12,8,1
74	4A	AH	0100 1010	$R1 \leftarrow c(R1)+c(S2)_{0-15}$	12,8,2
75	4B	SH	0100 1011	$R1 \leftarrow c(R1)-c(S2)_{0-15}$	12,8,3
76	4C	MH	0100 1100	$R1 \leftarrow (c(R1)Xc(S2)_{0-15})_{15-47}$	12,8,4
77	4D		0100 1101		12,8,5
78	4E	CVD	0100 1110	$S2_{0-63}$ (packed dec.) $\leftarrow$ c(R1)(binary)	12,8,6
79	4F	CVB	0100 1111	R1(binary) $\leftarrow c(S2)_{0-63}$ (packed dec.)	12,8,7
80	50	ST	0101 0000	$S2 \leftarrow c(R1)$	12
81	51		0101 0001		12,11,9,1
82	52		0101 0010		12,11,9,2
83	53		0101 0011		12,11,9,3
84	54	N	0101 0100	R1←c(R1) AND c(S2)	12,11,9,4
85	55	CL	0101 0101		12,11,9,5
86	56	O	0101 0110	R1←c(R1) OR c(S2)	12,11,9,6
87	57	X	0101 0111	R1←C(R1) Ex. OR c(S2)	12,11,9,7
88	58	L	0101 1000	R1←c(S2)	12,11,9,8
89	59	C	0101 1001		11,8,1
90	5A	A	0101 1010	R1←c(R1)+c(S2)	11,8,2
91	5B	S	0101 1011	R1←c(R1)−c(S2)	11,8,3
92	5C	M	0101 1100	[R1,R1+1]←c(R1+1)Xc(S2)	11,8,4
93	5D	D	0101 1101	R1←Rem. of [c(R1),c(R1+1)] /c(S2) R1+1←Quot. of [c(R1),c(R1+1)] /c(S2)	11,8,5
94	5E	AL	0101 1110	R1←c(R1)+c(S2)	11,8,6
95	5F	SL	0101 1111	R1←c(R1)−c(S2)	11,8,7
96	60	STD	0110 0000	S2←c(FPR1)	11
97	61		0110 0001		0,1
98	62		0110 0010		11,0,9,2
99	63		0110 0011		11,0,9,3
100	64		0110 0100		11,0,9,4
101	65		0110 0101		11,0,9,5
102	66		0110 0110		11,0,9,6
103	67		0110 0111		11,0,9,7
104	68	LD	0110 1000	FPR1←c(S2)	11,0,9,8
105	69	CD	0110 1001		0,8,1
106	6A	AD	0110 1010	FPR1←c(FPR1)+c(S2)	12,11
107	6B	SD	0110 1011	FPR1←c(FPR1)−c(S2)	0,8,3
108	6C	MD	0110 1100	FPR1←c(FPR1)Xc(S2)	0,8,4
109	6D	DD	0110 1101	FPR1←c(FPR1)/c(S2)	0,8,5
110	6E	AW	0110 1110	FPR1←c(FPR1)+c(S2)	0,8,6
111	6F	SW	0110 1111	FPR1←c(FPR1)−c(S2)	0,8,7
112	70	STE	0111 0000	S2←c(FPR1)	12,11,0
113	71		0111 0001		12,11,0,9,1
114	72		0111 0010		12,11,0,9,2
115	73		0111 0011		12,11,0,9,3
116	74		0111 0100		12,11,0,9,4
117	75		0111 0101		12,11,0,9,5
118	76		0111 0110		12,11,0,9,6
119	77		0111 0111		12,11,0,9,7
120	78	LE	0111 1000	FPR1←c(S2)	12,11,0,9,8
121	79	CE	0111 1001		8,1
122	7A	AE	0111 1010	FPR1←c(FPR1)+c(S2)	8,2
123	7B	SE	0111 1011	FPR1←c(FPR1)−c(S2)	8,3
124	7C	ME	0111 1100	FPR1←c(FPR1)Xc(S2)	8,4
125	7D	DE	0111 1101	FPR1←c(FPR1)/c(S2)	8,5
126	7E	AU	0111 1110	FPR1←c(FPR1)+c(S2)	8,6
127	7F	SU	0111 1111	FPR1←c(FPR1)−c(S2)	8,7

4 BYTE INSTRUCTIONS (RS,SI)

Decimal	Hexadecimal	Mnemonic	EBCDIC	Format	Instruction Action	EBCDIC Extended Hollerith EAM Card Code
128	80	IDL*	1000 0000	SI		12,0,8,1
129	81		1000 0001			12,0,1
130	82	PC*	1000 0010	SI		12,0,2
131	83	(DIG)*	1000 0011	SI		12,0,3
132	84	WRD*	1000 0100	SI		12,0,4
133	85	RDD*	1000 0101	SI		12,0,5
134	86	BXH	1000 0110	RS	R1←c(R1)+c(R3); if R3 is even→S2 if c(R1)≤c(R3+1) if R3 is odd→S2 if c(R1)≤c(R3)	12,0,6
135	87	BXLE	1000 0111	RS	R1←c(R1)+c(R3); if R3 is even→S2 if c(R1)>c(R3+1) if R3 is odd→S2 if c(R1)>c(R3)	12,0,7
136	88	SRL	1000 1000	RS	Right shift bits 0-31, fill 0's.	12,0,8
137	89	SLL	1000 1001	RS	Left shift bits 0-31, fill 0's.	12,0,9
138	8A	SRA	1000 1010	RS	Right shift bits 1-31, fill $c(R1)_0$.	12,0,8,2
139	8B	SLA	1000 1011	RS	Left shift bits 1-31, fill with (0).	12,0,8,3
140	8C	SRDL	1000 1100	RS	Right shift bits 0-63, fill 0's.	12,0,8,4
141	8D	SLDL	1000 1101	RS	Left shift bits 0-63, fill 0's.	12,0,8,5
142	8E	SRDA	1000 1110	RS	Right shift bits 1-63, fill $c(R1)_0$.	12,0,8,6
143	8F	SLDA	1000 1111	RS	Left shift bits 1-63, fill 0's.	12,0,8,7
144	90	STM	1001 0000	RS	S2...←c(R1),...,c(R3)	12,11,8,1
145	91	TM	1001 0001	SI		12,11,1
146	92	MVI	1001 0010	SI	$S1_{0-7}$←I2	12,11,2
147	93	TS	1001 0011	SI		12,11,3
148	94	NI	1001 0101	SI	$S1_{0-7}$←$c(S1)_{0-7}$ AND I2	12,11,4
149	95	CLI	1001 0101	SI		12,11,5
150	96	OI	1001 0110	SI	$S1_{0-7}$←$c(S1)_{0-7}$ OR I2	12,11,6
151	97	XI	1001 0111	SI	$S1_{0-7}$←$c(S1)_{0-7}$ Ex. OR I2	12,11,7
152	98	LM	1001 1000	RS	R1,...,R3←c(S2)...	12,11,8
153	99		1001 1001			12,11,9
154	9A		1001 1010			12,11,8,2
155	9B		1001 1011			12,11,8,3
156	9C	SDV*	1001 1100	SI		12,11,8,4
157	9D	TDV*	1001 1101	SI		12,11,8,5
158	9E	HDV*	1001 1110	SI		12,11,8,6
159	9F	CKC*	1001 1111	SI		12,11,8,7
160	A0		1010 0000			11,0,8,1
161	A1		1010 0001			11,0,1
162	A2		1010 0010			11,0,2
163	A3		1010 0011			11,0,3
164	A4		1010 0100			11,0,4
165	A5		1010 0101			11,0,5
166	A6		1010 0110			11,0,6
167	A7		1010 0111			11,0,7
168	A8		1010 1000			11,0,8
169	A9		1010 1001			11,0,9
170	AA		1010 1010			11,0,8,2
171	AB		1010 1011			11,0,8,3
172	AC		1010 1100			11,0,8,4
173	AD		1010 1101			11,0,8,5
174	AE		1010 1110			11,0,8,6
175	AF		1010 1111			11,0,8,7
176	B0		1011 0000			12,11,0,8,1
177	B1		1011 0001			12,11,0,1
178	B2		1011 0010			12,11,0,2
179	B3		1011 0011			12,11,0,3
180	B4		1011 0100			12,11,0,4
181	B5		1011 0101			12,11,0,5
182	B6		1011 0110			12,11,0,6
183	B7		1011 0111			12,11,0,7
184	B8		1011 1000			12,11,0,8
185	B9		1011 1001			12,11,0,9
186	BA		1011 1010			12,11,0,8,2
187	BB		1011 1011			12,11,0,8,3
188	BC		1011 1100			12,11,0,8,4
189	BD		1011 1101			12,11,0,8,5
190	BE		1011 1110			12,11,0,8,6
191	BF		1011 1111			12,11,0,8,7

6 BYTE INSTRUCTIONS (SS)

Decimal	Hexadecimal	Mnemonic	EBCDIC	Instruction Action	EBCDIC Extended Hollerith EAM Card Code
192	C0		1100 0000		12,0
193	C1		1100 0001		12,1
194	C2		1100 0010		12,2
195	C3		1100 0011		12,3
196	C4		1100 0100		12,4
197	C5		1100 0101		12,5
198	C6		1100 0110		12,6
199	C7		1100 0111		12,7
200	C8		1100 1000		12,8
201	C9		1100 1001		12,9
202	CA		1100 1010		12,0,9,8,2
203	CB		1100 1011		12,0,9,8,3
204	CC		1100 1100		12,0,9,8,4
205	CD		1100 1101		12,0,9,8,5
206	CE		1100 1110		12,0,9,8,6
207	CF		1100 1111		12,0,9,8,7
208	D0	SSP*	1101 0000		11,0
209	D1	MVN	1101 0001	S1←c(S2)	11,1
210	D2	MVC	1101 0010	S1←c(S2)	11,2
211	D3	MVZ	1101 0011	S1←c(S2)	11,3
212	D4	NC	1101 0100	S1←c(S1) AND c(S2)	11,4
213	D5	CLC	1101 0101		11,5
214	D6	OC	1101 0110	S1←c(S1) OR c(S2)	11,6
215	D7	XC	1101 0111	S1←c(S1) Ex. OR c(S2)	11,7
216	D8	LSP*	1101 1000		11,8
217	D9		1101 1001		11,9
218	DA		1101 1010		12,11,9,8,2
219	DB		1101 1011		12,11,9,8,3
220	DC	TR	1101 1100	S1←c(S2)	12,11,9,8,4
221	DD	TRT	1101 1101	Reg. 1_{8-31}←Address of Arg. byte; Reg. 2_{24-31}←Function byte	12,11,9,8,5
222	DE	ED	1101 1110	S1←c(S2)	12,11,9,8,6
223	DF	EDMK	1101 1111	S1←c(S2);Reg. 1_{8-31}←Addr. of 1st Sig. digit	12,11,9,8,7
224	E0		1110 0000		0,8,2
225	E1		1110 0001		11,0,9,1
226	E2		1110 0010		0,2
227	E3		1110 0011		0,3
228	E4		1110 0100		0,4
229	E5		1110 0101		0,5
230	E6		1110 0110		0,6
231	E7		1110 0111		0,7
232	E8		1110 1000		0,8
233	E9		1110 1001		0,9
234	EA		1110 1010		11,0,9,8,2
235	EB		1110 1011		11,0,9,8,3
236	EC		1110 1100		11,0,9,8,4
237	ED		1110 1101		11,0,9,8,5
238	EE		1110 1110		11,0,9,8,6
239	EF		1110 1111		11,0,9,8,7
240	F0		1111 0000		0
241	F1	MVO	1111 0001	S1←c(S2)	1
242	F2	PACK	1111 0010	S1(packed dec.)←c(S2) (zoned dec.)	2
243	F3	UNPK	1111 0011	S1(zoned dec.)←c(S2) (packed dec.)	3
244	F4		1111 0100		4
245	F5		1111 0101		5
246	F6		1111 0110		6
247	F7		1111 0111		7
248	F8	ZAP	1111 1000	S1←c(S2)	8
249	F9	CP	1111 1001		9
250	FA	AP	1111 1010	S1←c(S1)+c(S2)	12,11,0,9,8,2
251	FB	SP	1111 1011	S1←c(S1)−c(S2)	12,11,0,9,8,3
252	FC	MP	1111 1100		12,11,0,9,8,4
253	FD	DP	1111 1101		12,11,0,9,8,5
254	FE		1111 1110		12,11,0,9,8,6
255	FF		1111 1111		12,11,0,9,8,7

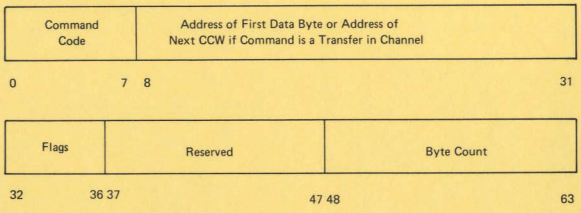
SUMMARY OF CONSTANTS

Code	Type	Alignment	Implied Length (Bytes)	Explicit Length Modifier Range (LM)	Truncation Padding Side	Constant Specified by	Number of Constants Per Statement
A	Address	word(1)	4	1 to 4	left	any expression	one
B	Binary	byte	as needed	1 to 256	left	binary digits	one
C	Character	byte	as needed	1 to 256	right	characters	one
D	Long Float. Pt.	double word(1)	8	1 to 8	right	decimal digits	multiple
E	Short Float. Pt.	word(1)	4	1 to 8	right	decimal digits	multiple
F	Fixed Point	word(1)	4	1 to 8	left	decimal digits	multiple
H	Fixed Point	half word (1)	2	1 to 8	left	decimal digits	multiple
P	Packed Decimal	byte	as needed	1 to 16	left	decimal digits	multiple
S	Base And Displacement Form of Address	half word(1)	2	2 only	left	Absolute Expression for Base Any Expression for Displacement	one
V	External Branch Pt. Address	word(1)	4	3 or 4	left	relocatable symbol	one
X	Hexadecimal	byte	as needed	1 to 256	left	hexadecimal digits	one
Y	Half Word Address (≤65K)	half word(1)	2	1 to 4	left	any expression	one
Z	Zoned Decimal	byte	as needed	1 to 16	left	decimal digits	multiple

NOTE:

1 – Alignment given applies to impled length only.

CHANNEL COMMAND WORD



COMMAND CODE OPERATIONS

Command Code								Operation
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	Bit Position
M	M	M	M	0	0	0	1	Sense
M	M	M	M	M/0	0	1	0	Read Reverse
M	M	M	M/B	M/0	0	1	1	Write
M	M	M	M/B	M/0	1	0	0	Write Erase
M	M	M	M/B	M/0	1	0	1	Read
M	M	M	M	0	1	1	1	Write Control
M	M	M	M	1	0	0	1	Transfer in Channel

ASSEMBLER DIRECTIVES FORMAT SUMMARY

DROP	r1, r2, . . . ,rn
DSECT	
EJECT	
END	[entry-point]
ENTRY	symbolic-name
EQU	expression
EXTRN	symbolic-name
ICTL	b[,e] [,c]
ISEQ	[L,R]
PRINT	[{SINGLE / DOUBLE}] [,{ON / OFF}] [,{GEN / NOGEN}] [,{ION / IOFF}] [,{DATA / NODATA}] [,{DECK / NO DECK}] [,{NUM / NONUM}] [,{OPEN / CLOSED}]
PUNCH	'up to 80 characters to be punched'
REPRO	
SPACE	[n]
START	
TITLE	'up to 100 characters'
USING	symbol, r1, r2, . . . ,rn