UC Reference Summary

IBM CONFIDENTIAL

DO NOT REPRODUCE Pages Marked IBM CONFIDENTIAL

SUMMARY OF UC PROGRAMMING SUPPORT TOOLS FOR VM/CMS ENVIRONMENTS

UC ASSEMBLER

UCASM fname [options]

options: [LIB] [NOLIB] [NOGLB] [PUNCH] [BRIEF] [ERMES] [EDIT] [PRT] [LIST] [NOOPT] [OPT] [Cn]

NUCASM fname [ftype [fmode]] [(options]

options: [ALOGIC] [NOALOGIC] [RLD] [NORLD] [NUM] [NONUM]

[FLAG nnn] [ESD] [NOESD] [XREF] [NOXREF] [XREF SHORT] [STMT] [NOSTMT]

[LINECOUNT nn]

[SUBSET subsetname] [LIST] [NOLIST] [DECK] [NODECK] [TERM] [NOTERM]

[MCALL] [NOMCALL] [OBJ] [NOOBJ] [ALIGN] [NOALIGN]

[MLOGIC] [NOMLOGIC] [TEST] [NOTEST]

[LIBMAC] [NOLIBMAC]

UC LINKAGE EDITOR

UCLINK fname [options]

options: [SYM] [NOSYM] [CLEAR0] [EPOINT]

[ESD] [NOESD] [COL71] [ID]

[TYPE] [NOTYPE] [DOCUMENT] [Knnn]

[XREF] [NOXREF]

Allowable UCLINK Statements:

MAP OVER csectname
AT hex-location
REL

FILE modulefname textfname [(alignf)] [(CSECT csectlist)]
. . . textfname [(alignf)] [(CSECT csectlist)]

where csectlist is
csectname [(alignf)] . . . csectname [(alignf)]

CONTINUE textfname [(alignf)] [(CSECT csectlist)] . . .
textfname [(alignf)] [(CSECT csectlist)]

SLC hexloc

SLCB hexloc

OVERLAY hexloc hexloc

LIBSRCH ftype1 [ftype2] . . . [ftype8]

CHANGE textfname oesd=nesd . . . . oesd=nesd

* comments

TEXTDICT – Text Dictionary Routine

TEXTDICT GEN $dictfn textfnl . . . textfnx
TEXTDICT ADD $dictfn textfnl . . . textfnx
TEXTDICT DEL $dictfn textfnl . . . textfnx
TEXTDICT LIST $dictfn

PROGRAM DEBUG SIMULATOR INVOCATION

PDSCL

PDSCL Commands

- UC UCMODEL = UC.0 , RSZ=r, MSZ={m / nk} [,UCNUM=x]
  UC.5
  [,MSZ1={m / nk}] [,ROS= addr/addr / FFFF].

- ADAPTER DV=device, AD=address, LV=level, IP={T / N / D} , OP={T / P / D}
  [,other].

- LOAD membername [,membername . . .].

- SET facilityname=value [,facilityname=value . . .].

  facilityname: CLOCK, IOIRR, IC, MMB, TIMEOUT, LONGMESS, CMNMSK, PIRR, MCPC, OLNCNT, IDONE, UCNUM, PRYPGE, SDYPGE, RpPnn, ICOUNT, WINDOW, PSW0–PSW7

- MODIFY addr=value [,addr=value . . .].

- DISPLAY facilityname
  MAIN=addr/addr [, . . .].
  REGS
  label

  facilityname: CLOCK, IOIRR, IC, MMB, PIC, ZHCV, IOBUS, UCNUM, CMNMSK, PIRR, MCPC, LEVELS, UCMODEL, CONDS, OPCODE, LONGMESS, PRYPGE, SDYPGE, MSZ, RSZ, RPnn, INST, PSW0–PSW7

- WAIT.

- IOINT ± n.

- MAP {B / E / C / S / L}

- GO.

- END.

- NAME labelx.
  KEEP.
  RETRIEVE labely[, . . .] , # L.
  , # LO
  CLEAR labely.

- TRACE. [only for conditional sentences]

- PAUSE [only for conditional sentences]

- STAT (B / C / D / E)

PDSCL STATEMENT FORMATS

Sentence format:

command1; command2; . . . . commandn.

Conditional Sentence format:

label: [ON] cond1,cond2, . . . ,condn command1; . . . ,commandn.

conditions:

MC	L=n	INT	PIC = low / high
PC	IC=low/high	ERA=xx	WAIT
Ln	BR=low/high	CSG	EWAIT

UC MACHINE INSTRUCTIONS

	Instruction	Format	Mnemonic	Operands	Op+Extn
X	Add Register Immediate	RI	ARI	P,I8	3
*	Add Halfword Register Immediate	RRH	AHRI	H,I4	C D
	Add Register	RR	AR	PSP,PSP	7 8,9x
*	Add Halfword Register	RRH	AHR	H,H	C 8
	Add with Carry Register	RR	AYR	PSP,PSP	7 A,Bx
*	Add with Carry Halfword Register	RRH	AYHR	H,H	C A
	And Register Immediate	RI	NRI	P,I8	0
	And Register	RR	NR	PSP,PSP	7 0,1x
*	And Halfword Register	RRH	NHR	H,H	C 0
X	Branch on Condition Register	RRH	BCR	MASK,H	A 0
	Branch and Link Register	RRH	BALR	H,H	A 3
*	Branch and Count Register	RRH	BCTR	P,H	A 2
	Compare Register	RR	CR	PSP,PSP	7 6,7x
*	Compare Halfword Register	RRH	CHR	H,H	C 6
	Control Immediate	RI	KI	PE,I8	6
*	Count Leading Zeros	RRH	CTLZ	H,H	C 1
	Exclusive Or Register Immediate	RI	XRI	P,I8	2
	Exclusive Or Register	RR	XR	PSP,PSP	7 4,5x
*	Exclusive Or Halfword Register	RRH	XHR	H,H	C 4
	I/O Immediate	RI	IOI	PO,I8	6
	I/O	RR	IO	P,H	A 4
*	I/O Halfword	RRH	IOH	H,H	A 5
X	Jump on Condition	RI	JC	MASK,S	9 0y
*	Jump on Bit Zero	RI	JBZ	BIT,S	9 1y
	Load Register Immediate	RI	LRI	P,I8	4
	Load Indirect	RR	LN	P,H	A B
*	Load Halfword Indirect	RRH	LHN	H,H	A 9
*	Load Halfword Short	RS	LHS	H,AS	B 1y
	Load Register Space Indirect	RR	LRN	P,H	A F
*	Load Hwd Reg Space Indirect	RRH	LHRN	H,H	A D
	Load Register	RR	LR	RA,RA	8 4–7x
*	Load Halfword Register	RRH	LHR	H,H	C 3
	Or Register Immediate	RI	ORI	P,I8	1
	Or Register	RR	OR	PSP,PSP	7 2,3x
*	Or Halfword Register	RRH	OHR	H,H	C 2
	Rotate Left	RR	RL	RA,I3	8 2,3x
*	Rotate Left Halfword	RRH	RLH	H,I4	C B
	Shift Left Logical	RR	SLL	RA,I3	8 0,1x
*	Shift Left Halfword Logical	RRH	SLHL	H,I4	C 9
	Store Indirect	RR	STN	P,H	A 8
*	Store Halfword Indirect	RRH	STHN	H,H	A A
*	Store Halfword Short	RS	STHS	H,AS	B 0y
	Store Register Space Indirect	RR	STRN	P,H	A C
*	Store Hwd Reg Space Indirect	RRH	STHRN	H,H	A E
*	Test and Set	RRH	TS	0,H	A 1
	Subtract Register	RR	SR	PSP,PSP	7 C,Dx
*	Subtract Halfword Register	RRH	SHR	H,H	C C
	Subtract with Carry Register	RR	SYR	PSP,PSP	7 E,Fx
*	Subtract with Carry Hwd Register	RRH	SYHR	H,H	C E
*	Subtract Halfword Reg Immediate	RRH	SHRI	H,I4	C F
	Test Register Immediate	RI	TRI	P,I8	5

* = Not available on UC 0 X = Extended mnemonics available for this instruction

UNIVERSAL CONTROLLER CODE POINT ASSIGNMENTS

	0	3 4 5	7 8	11 12	14 15	CC
NRI	0	P	I8			L
ORI	1					L
XRI	2					L
ARI	3					A
LRI	4					
TRI	5					T
IOI	6	PO				I
KI		PE				

	0	3 4 5	7 8	11 12	PP SP	CC
NR	7	PSP	PSP		0 / 1	L
OR					2 / 3	L
XR					4 / 5	L
CR					6 / 7	C
AR					8 / 9	A
AYR					A / B	A
SR					C / D	A
SYR					E / F	A

	0	3 4 5	7 8	11 12	14 15	CC
SLL	8	0 C3	RA (PP)	0		S
SLL			(SP)	1		S
RL			(PP)	2		S
RL			(SP)	3		S
LR		RA (PP)	(PP)	4		
LR		(PP)	(SP)	5		
LR		RA (SP)	(PP)	6		
LR		(SP)	(SP)	7		

	0	3 4 5	7 8	11 12	14 15	CC
JC	9	MASK	I8		0	
JBZ		BIT			1	

	0	3 4 5	7 8	11 12	14 15	CC
BCR	A	MASK	H	0		
TS		0		1		L
BCTR		P		2		
BALR		H		3		
IO		P		4		I
IOH		H		5		I
STN		P		8		
LHN		H		9		
STHN		H		A		
LN		P		B		
STRN		P		C		
LHRN		H		D		
STHRN		H		E		
LRN		P		F		

	0	3 4	7 8	11 12	15	CC
NHR	C	H	H	0		L
OHR				2		L
XRH				4		L
CHR				6		C
AHR				8		A
AYHR				A		A
SHR				C		A
SYHR				E		A
CTLZ				1		L
LHR				3		
SLHL		C4		9		S
RLH				B		S
AHRI		I4		D		A
SHRI				F		A

	0	3 4	7 8	11 12	15
STHS	B	H	B	DH	0
LHS					1

Symbol Key

P	Primary page byte register, value 0 to 15
PE	Even primary page byte register, value 0 to 15 (even)
PO	Odd primary page byte register, value 0 to 15 (odd)
PSP	PSP,PSP indicates byte registers in either primary or secondary page, but both operand registers must be in the same page
RA	Any register, value 0 to 31
PP	Primary page
SP	Secondary page
MASK	Mask, value 0 to 15 (4 bits)
I4	Immediate field, value 0 to 15
I8	Immediate field, value 0 to 255
C3	Count field, value 0 to 7
C4	Count field, value 0 to 15
B	Base register specification, 12(b'00'), 14(b'01'), 28(b'10'), 30(b'11')
BIT	Bit to be tested in halfword register 2
H	Halfword register, value 0 to 30 (even)
DH	Displacement in halfwords
AS	Either SA or DS or DS(BS)
SA	Relocatable or absolute expression. SA minus the value of the contents in the base register used should be in the range 0 to 62
DS	Displacement, value 0 to 62 (even)
BS	Base register, value 12, 14, 28, 30
S	Relocatable expression in Jump instructions and relocatable or absolute expression in GOTO instructions. In Jump instructions the value of S subtracted by the location counter value of the instruction following the Jump instruction should be in the range −128 to +126
x	Bit 15 of the instruction indicates that the register page used is 0 – primary 1 – secondary
Y	Bit 15 is used to distinguish operations
A	Arithmetic condition code set
L	Logical condition code set
I	I/O condition code set
S	Shift condition code set
T	Test condition code set

INSTRUCTION TIMINGS

INSTRUCTION	UC.5
ARI	2.8
AHRI	2.8 + (1.2) c
AR	3.6
AHR	4.8
AYR	3.6
AYHR	4.8
NRI	2.4
NR	3.2
NHR	3.2
BCR	3.6 (B) / 1.6 (NSI)
BALR	2.8 (B) / 1.6 (NSI)
BCTR	3.6 (B) / 3.2 (NSI)
CR	3.6
CHR	4.8
KI	2.8–5.2 *
CTLZ	4.8
XRI	2.4
XR	3.2
XHR	3.2
IOI (read)	6.4 (B) / 5.6 (H)
(write)	7.2 (B) / 6.4 (H)
IO (read)	6.0 (B) / 5.2 (H)
(write)	6.8 (B) / 6.0 (H)
IOH(read)	– – – / 4.8 (H)
(write)	– – – / 5.6 (H)
JC	2.4 (J) / 1.6 (NSI)
JBZ	3.2 (J) / 2.4 (NSI)
LRI	1.6
LN	3.2
LHN	3.2
LHS	4.0 + (1.2) c
LRN	3.2
LHRN	3.2
LR	2.4
LHR	2.4
ORI	2.4
OR	3.2
OHR	3.2
RL	1.6 + (1.2*n) / 2.4 (n=0)
RLH	1.6 + (1.2*n) / 2.4 (n=0)
SLL	1.6 + (1.2*n) / 2.4 (n=0)
SLHL	1.6 + (1.2*n) / 2.4 (n=0)
STN	3.2
STHN	3.2
STHS	3.6 + (1.2) c
STRN	3.2
STHRN	3.2
SR	3.6
SHR	4.8
SYR	3.6
SYHR	4.8
SHRI	2.8 + (1.2) c
TRI	2.4
TS	4.0
PSW swap	6.4
CS read	5.6 + 1.6*t
CS write	6.4 + 1.6*t

Cut along dotted line

LPL – MICROCODE DEVELOPMENT SYSTEM

- LDEFLVL level [level1 . . . levelN] [(options]

  To define a new level, update an existing level definition or change the write lock on a level.

  options: [NEW / UPD] [*NGBL / NONGBL / NGBL] [*SGBL / NOSGBL / SGBL] [A1 / B1 / C1 / D1 / E1 / F1 / G1]
  [NOBASE / BASE] [*AGBL / NOAGBL / AGBL] [*LGBL / NOLGBL / LGBL]
  [NOPROM / PROM] [*TGBL / NOTGBL / TGBL] [*PGBL / NOPGBL / PGBL]
  [WRITE / NOWRITE]

- LDEL level fname ftype [TSfname TSftype] [(options]

  To delete a file from the library.

  options: [DEL / PROMD]

- LDUMP level [(options]

  To dump all the files of a level to tape, or selected ones.

  options: [ALL / MACS / NOTMACS / SYSTEM / USER / CHARTS / OBJECT / TOOLS] [ML / NOML] [CTL / NOCTL] [PRINT / NOPRINT]

- LGET level fname ftype [AS fname [ftype [fmode]] [fmode]

  To obtain a data file from the library.

- LLF fname [ftype [fmode]] [(options]

  options: [NAME / TYPE / MODE / FORMAT / ALLOC / DATE] [OFF] [ADD]

- LLIST level [subset] [(options]

  To list the contents of a level.

  subset: [MACS / NOTMACS / SYSTEM / USER / CHARTS / OBJECT / LCMS / ALL / fname ftype]

  Options: [TERM / PRINT] [NOFILE / FILE]

  Note: fname may be *
  ftype may be *
  ftype default is SYSIN

- LMARK level fname ftype [FROM TSfname TSftype TSfmode] [(options]

  To specify reprocessing of a library file when the file itself has not been modified.

  options: [NORDR / RDR] [LMARK / LREAD] [PRM nnnnnnnn]

- LPROM level fname ftype [FROM TSfname TSftype TSfmode] [(options]

  To promote a file one level higher in its level chain

  options: [NORDR / RDR] [LPROM / LREAD] [NOPROMPT / PROMPT] [CHECK / NOCHECK]
  [PRM nnnnnnnn]

- LPUT level fname ftype [FROM fname [ftype [fmode]] / fmode] [(options]

  To return an updated data file to the library.

  options: [NORDR / RDR] [LPUT / LREAD] [NOPROMPT / PROMPT] [CHECK / NOCHECK]
  [PRM nnnnnnnn]

- LREAD [options]

  To read files from the input reader and update the library.

  options: [CHECK / NOCHECK] [NOPROMPT / PROMPT] [HOLDS] [NOATTN / ATTN / ATTN1]

- LVER level fname ftype [fmode]

  May be used to determine if the top time stamp of the user file has been received by the library level.

- LASM level fname [ftype] [(options]

  To assemble a UC assembler language or 370 assembler language program using NUCASM or HASM respectively.

  options: [PE / NOPE] [LIST / NOLIST / NL] [DECK / NODECK / ND]
  [XREF / NOXREF / NX] [NOTERM / TERM / T] [RLD / NORLD / R]
  [SYM / NOSYM] [PRINT / NOPRINT] [ERASE / NOERASE]
  [NOPUNCH / PUNCH] [B1 / A1] [PROMPT / GO / QUIT]
  [NOLIB / LIB] [LIBSRCH / LIBHIST / NOSEARCH]

  ftype possibilities:

  SYSIN
  APB
  CPGEN
  ASSEMBLE

- LLINK level fname [(options]

  To link-edit text files into a UC module.

  Options: [PRINT / NOPRINT] [ERASE / NOERASE] [NOPUNCH / PUNCH]

  [XREF / NOXREF / NX] [NOTYPE / TYPE / T] [ESD / NOESD / EPOINT]

  [SYM / NOSYM] [CLEAR0 / CLEARF] [TEXT / NOTEXT]

  [NOLIB / LIB] [B1 / A1] [K64 / K128 / K32]

- LMAC level fname [ftype] [(options]

  To insert, replace or delete one macro or copy member of a maclib.

  Options: [TS / NOTS] [NOSEARCH / SEARCH] [NOLIB / LIB] [DEL / ADD / GEN]

- LMAK level [function / fname ftype] [(options]

  To update a library maclib file.

  Function: LCMS

  CHANGES [ALL / SML / AML / PML]

  Options: [NOSEARCH / SEARCH] [NOPRINT / PRINT]

  [NOCOMP / COMP] [DEL / GEN / ADD]

filetype possibilities	MACLIB
MACRO COPY	SML$LVL
APBMAC APBCPY	AML$LVL
INCLUDE	PML$LVL

- LLCTL level [(options]

  Types, prints or creates a file containing the contents of the level definition file for a level.

  Options: [TYPE / PRINT / FILE] [NOCHANGE / CHANGE]

- LLIB dmode [BYTYPE / BYNAME]

  Prints a sorted list of all files on a disk.

- LLEXEC dmode [EXEC / EDMACRO]

  To list the execs and edmacros on a disk.

- LSTAT

  To list the status of the library reader and the outstanding updates by $LIB file in the library.

- LTOUT execname [ALL / OFF]

  Used to debug the library execs by making a copy of an exec on the A-disk which has all exec printing turned on or OFF as appropriate.

- LDELTS fname ftype fmode [NOTALL / ALL]

  To delete the time stamps from a file.

- LMSG ufname message

  Broadcast a message to a group of users.

  ufname: [USERS / fname] (filetype must be $USERS)

  message: up to 16 words

- LMOD level [function] [(options]

  To assemble a library date file and place the results in the library.

  Function: CHANGES [ftype] / LCMS / fname [ftype]

  Options: [NOOKERR / OKERR] [NOIBMC / IBMC] [LIBSRCH / LIBHIST / NOSEARCH] [TL / NTL]

  ftype possibilities: SYSIN, APB, CPGEN, PLS2, ASSEMBLE, ALL

- LSORT fname ftype majb maje [minb1 mine1 . . . minb4 . . . mine4]

  To sort a file.

- Library/CMS filetypes
  CCC$ level

LIBRARY (CCC)	CMS
SRC	SYSIN
APB	APB
CPG	CPGEN
MAC	MACRO
CPY	COPY
AMF	APBMAC
ACF	APBCOPY
FL1	FL1
UCL	UCLINK
UCM	UCMOD
CRC	CONTROL
MCI	MCITIN
TXT	TEXT
PLS	PLS2
XEC	EXEC
JCL	JCL
UPD	UPDATE
MOD	MODULE
KRT	KEEPRET
LNK	LINK
MAP	MAP
EDM	EDMACRO
ASM	ASSEMBLE
LST	LISTING
CLT	CLIST
SCT	SCRIPT
INC	INCLUDE

- Time Stamp Format

Column	Description
1–8	ID & Comment Notation
9–12	Level
14–21	FILENAME
23–30	FILETYPE
32–39	DATE
41–48	TIME
50–57	USERID
59–62	FROM level (promote only)
64–71	USER FIELD

Notes

Notes

m = number of passes; 1 pass = shift of 1, 2, 4, or 8 bits
t = number of transfers (bytes or halfwords)
c = added if carry occurred

** KI – Read/Write Page Pointers	2.8/2.8
KI – Read/Write MC/PC Status	2.8/5.2
Read/Write PSC	2.8/2.8
Reset/Set Master Mask	4.0/4.0
Read/Write Common Mask	3.6/4.0
Read/OR/AND PIRR	3.6/4.0/4.0
Write Next Level	4.0
Read I/O Interrupts	3.6
Read/Write Levels	3.6/4.8

EXTENDED MNEMONICS

Explanation	*Extended*		*Standard*	
Subtract reg. immed.	SRI	P,I	ARI	P,–I

EXTENDED BRANCH MNEMONIC OPERATIONS

Explanation	*Extended*		*Standard*		*Mask*
After arithmetic instr.					
Branch if zero	BZR	H	BCR	8,H	1000
Branch if plus	BPR	H	BCR	2,H	0010
Branch if minus	BMR	H	BCR	4,H	0100
Branch if not zero	BNZR	H	BCR	6,H	0110
Branch if not plus	BNPR	H	BCR	12,H	1100
Branch if not minus	BNMR	H	BCR	10,H	1010
Branch if carry	BYR	H	BCR	0,H	0000
Branch if overflow	BVR	H	BCR	1,H	0001
After compare instr.					
Branch if equal	BER	H	BCR	8,H	1000
Branch if high	BHR	H	BCR	2,H	0010
Branch if low	BLR	H	BCR	4,H	0100
Branch if not equal	BNER	H	BCR	6,H	0110
Branch if not high	BNHR	H	BCR	12,H	1100
Branch if not low	BNLR	H	BCR	10,H	1010
After logical instr.					
Branch if all zeros	BZR	H	BCR	8,H	1000
Branch if all ones	BOR	H	BCR	4,H	0100
Branch if mixed	BXR	H	BCR	2,H	0010
Branch if not all zeros	BNZR	H	BCR	6,H	0110
Branch if not all ones	BNOR	H	BCR	10,H	1010
Branch if not mixed	BNXR	H	BCR	12,H	1100
After test instr.					
Branch if all zeros U.M.	BZR	H	BCR	8,H	1000
Branch if all ones U.M.	BOR	H	BCR	4,H	0100
Branch if mixed U.M.	BXR	H	BCR	2,H	0010
Branch if not all zeros U.M.	BNZR	H	BCR	6,H	0110
Branch if not all ones U.M.	BNOR	H	BCR	10,H	1010
Branch if not mixed U.M.	BNXR	H	BCR	12,H	1100
Branch if equal to Mask	BTER	H	BCR	1,H	0001
After shift instr.					
Branch if all zeros	BZR	H	BCR	8,H	1000
Branch if hi-bit off	BPR	H	BCR	2,H	0010
Branch if hi-bit on	BMR	H	BCR	4,H	0100
Branch if not all zeroes	BNZR	H	BCR	6,H	0110
Branch if not positive	BNPR	H	BCR	12,H	1100
Branch if not negative	BNMR	H	BCR	10,H	1010
Branch if one bit moved out	BVR	H	BCR	1,H	0001
After any instruction					
Branch unconditionally	BR	H	BCR	15,H	1111

EXTENDED JUMP MNEMONIC OPERATIONS

Explanation	*Extended*		*Standard*		*Mask*
After arithmetic instr.					
Jump if zero	JZ	label	JC	8,label	1000
Jump if plus	JP	label	JC	2,label	0010
Jump if minus	JM	label	JC	4,label	0100
Jump if not zero	JNZ	label	JC	6,label	0110
Jump if not plus	JNP	label	JC	12,label	1100
Jump if not minus	JNM	label	JC	10,label	1010
Jump if carry	JY	label	JC	0,label	0000
Jump if overflow	JV	label	JC	1,label	0001
After compare instr.					
Jump if equal	JE	label	JC	8,label	1000
Jump if high	JH	label	JC	2,label	0010
Jump if low	JL	label	JC	4,label	0100
Jump if not equal	JNE	label	JC	6,label	0110
Jump if not high	JNH	label	JC	12,label	1100
Jump if not low	JNL	label	JC	10,label	1010
After logical instructions					
Jump if all zeros	JZ	label	JC	8,label	1000
Jump if all ones	JO	label	JC	4,label	0100
Jump if mixed	JX	label	JC	2,label	0010
Jump if not all zeros	JNZ	label	JC	6,label	0110
Jump if not all ones	JNO	label	JC	10,label	1010
Jump if not mixed	JNX	label	JC	12,label	1100
After test instructions					
Jump if all zeros U.M.	JZ	label	JC	8,label	1000
Jump if all ones U.M.	JO	label	JC	4,label	0100
Jump if mixed U.M.	JX	label	JC	2,label	0010
Jump if not all zeros U.M.	JNZ	label	JC	6,label	0110
Jump if not all ones U.M.	JNO	label	JC	10,label	1010
Jump if not mixed U.M.	JNX	label	JC	12,label	1100
Jump if equal to Mask	JTE	label	JC	1,label	0001
After shift instructions					
Jump if all zeros	JZ	label	JC	8,label	1000
Jump if hi-bit off	JP	label	JC	2,label	0010
Jump if hi-bit on	JM	label	JC	4,label	0100
Jump if not all zeros	JNZ	label	JC	6,label	0110
Jump if not positive	JNP	label	JC	12,label	1100
Jump if not negative	JNM	label	JC	10,label	1010
Jump if one bit moved out	JV	label	JC	1,label	0001
After any instruction					
Jump unconditionally	J	label	JC	15,label	1111

GOTO MNEMONICS

Expands to either a JUMP – JC MASK,S — If the target is sufficiently close to the GOTO instruction

or to the three instructions: – LRI &SYSSCRT,K(S)
LRI &SYSSCRT+1,L(S) — Load address into e/o pair of registers
BCR MASK,&SYSSCRT

The default setting of &SYSSCRT is 10 but can be changed via a SETA.

Extended Code	*Meaning*	*Mask*
After Arithmetic Instructions		
GZ	Goto if Zero	1000
GP	Goto if Plus	0010
GM	Goto if Minus	0100
GNZ	Goto if not Zero	0110
GNP	Goto if not Plus	1100
GNM	Goto if not Minus	1010
GY	Goto if Carry	0000
GV	Goto if Overflow	0001
After Compare Instructions		
GE	Goto if Equal	1000
GH	Goto if High	0010
GL	Goto if Low	0100
GNE	Goto if Not Equal	0110
GNH	Goto if Not High	1100
GNL	Goto if Not Low	1010
After Logical Instructions		
GZ	Goto if All Zeros	1000
GO	Goto if All Ones	0100
GX	Goto if Mixed Zeros and Ones	0010
GNZ	Goto if Not All Zeros	0110
GNO	Goto if Not All Ones	1010
GNX	Goto if Not Mixed	1100
After Test Instructions		
GZ	Goto if All Zeros Under Mask	1000
GO	Goto if All Ones Under Mask	0100
GX	Goto if Mixed Under Mask	0010
GNZ	Goto if Not All Zeros Under Mask	0110
GNO	Goto if Not All Ones Under Mask	1010
GNX	Goto if Not Mixed Under Mask	1100
GTE	Goto if Equal to Mask	0001
After Shift and Rotate Instructions		
GZ	Goto if All Zeros	1000
GP	Goto if Hi-bit Off (Not Zero)	0010
GM	Goto if Hi-bit On	0100
GNZ	Goto if Not All Zeros	0110
GNP	Goto if Not Positive	1100
GNM	Goto if Not Negative	1010
GV	Goto if Any One Bits Moved Out	0001
After Any Instruction		
G	Unconditional Goto	1111

Note: All GOTO instructions are coded as Gxx label

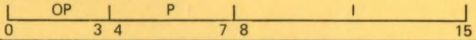
INSTRUCTION SEQUENCE CONTROL – JUMPS and BRANCHES

OPERATION	*CONDITION*	*TIME μs.*	*ACTIONS*
JC MASK,I8	TAKEN:	2.4	IC ← IC2+I8
	NOT TAKEN:	1.6	IC ← IC2
			(−128 ≤ I8 ≤ +126)
JBZ BIT,I8	P2 (BIT) = 0	3.2	IC ← IC2+I8
	P2 (BIT) ≠ 0	2.4	IC ← IC2
			(−128 ≤ I8 ≤ +126)
BCR MASK,H	TAKEN:	3.6	IC ← H
	NOT TAKEN:	1.6	IC ← IC2
BCTR P,H			P ← P−1
	P ≠ 0	3.6	IC ← H
	P = 0	3.2	IC ← IC2
BALR H1, H2	H2 ≠ 0	2.8	IC ← H2
			H1 ← IC2
	H2 = 0	1.6	IC ← IC2
			H1 ← IC2
	H1 = H2 = H	2.8	IC ← H2
			H1 ← IC2

Notes:
* IC2 = Address of the branch instruction plus two.
* P2 (BIT) = Bit in halfword register 2 addressed by BIT.

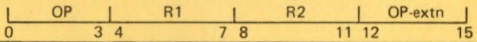
INSTRUCTION FORMATS

RI REGISTER IMMEDIATE

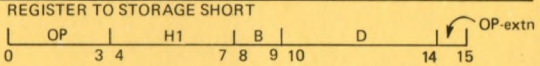


RR REGISTER TO REGISTER (byte register operands)

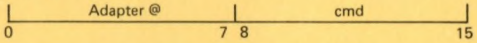
RRh REGISTER TO REGISTER (halfword register operands)



RS REGISTER TO STORAGE SHORT



I/O SPACE Address Format



Bits	Explanation
0–7	Adapter address
8–15	Command
15	0 – Write 1 – Read

REGISTER SPACE Address Format

Bits	Explanation
0–11	Page number
12–15	Register number within page

ASSIGNED CONTROL ADDRESSES

NO	OPERATION	NO	OPERATION
0	Disable Master Mask	1	unassigned
2	Write Common Mask	3	Read Common Mask
4	OR to PIRR	5	Read PIRR
6	AND to PIRR	7	Read Interrupt Requests (IOBI)
8	Write to MC/PC Status	9	Read MC/PC Status
10	Write Primary Page No.	11	Read Primary Page No.
12	Write Secondary Page No.	13	Read Secondary Page No.
14	Enable Master Mask	15	Read Levels (CLL & LLL)
16	unassigned	17	unassigned
18	unassigned	19	unassigned
20	unassigned	21	unassigned
22	unassigned	23	unassigned
24	unassigned	25	unassigned
26	Write PSC bits	27	Read PSC bits
28	Write next level	29	unassigned
30	unassigned	31	unassigned

Only addresses 0 to 15 are implemented on the UC 0.

Contents of first operand register (where meaningful).

Interrupt Sources:

BIT	PIRR & IOBI	MC/PC
0	Int. level 0	I/O parity check
1	Int. level 1	VALID line not raised
2	Int. level 2	Storage parity check
3	Int. level 3	Invalid operation
4	Int. level 4	Error during CS
5	Int. level 5	Reserved
6	Int. level 6	Modifier Latch
7	Int. level 7	Instruction length code

Reading Levels:
Bits 0–3 Present Level Latch
Bits 4–7 Last Level Latch

Page Numbers:
Bits 0–1 00
Bits 2–7 number

PSC Bits:
Bits 0–3 0000
Bits 4–7 ZHCV

SELF DEFINING CONSTANTS

CODE	*TYPE*	*MEANING*
C	Character	8 bit EBCDIC code for each character
X	Hexidecimal	4 bit hexidecimal code for each digit
B	Binary	Binary digits
n	Decimal	n = the decimal value wanted appropriate binary value used
K	Address	Block portion of 16 bit address (bits 0–7)
L	Address	Displacement portion of address (bits 8–15)

DEFINING CONSTANTS

CODE	*TYPE*	*MEANING*
C	Character	8 bit EBCDIC code for each character
X	Hexidecimal	4 bit code for each hex digit
B	Binary	Binary digits
F	Fixed-point	Signed, fixed point format; normally a fullword
H	Fixed-point	Signed, fixed-point format; normally a halfword
I	Fixed-point	Signed, fixed-point format; normally a byte
A	Address	Value of address; normally a fullword
K	Address	Value of BLOCK portion of an address; one byte
L	Address	Value of displacement portion of an address; one byte
V	Address	Space reserved for ext. symbol; normally a fullword
Y	Address	Value of address; normally a halfword

CONDITION CODE SETTING DEFINITION

PSC BIT SETTINGS	(Z)	(H • $\bar{Z}$)	($\bar{H}$ • $\bar{Z}$)	(V)	(C)
CONDITION CODE	CC0	CC1	CC2	CC3	CCcry
MASK BIT POSITION	8	4	2	1	
MASK BIT PATTERN	B'1xxx'	B'x1xx'	B'xx1x'	B'xxx1'	B'0000'
A [ARITHMETIC]	=0	<0	>0	Ov'flow	Carry
C [COMPARE] (Arith.)	op1=op2	op1<op2	op1>op2	Ov'flow	Carry
(Logical)	op1=op2				op1≥op2
L [LOGICAL]	All 0's	All 1's	Mixed	Never Set	Never Set
T [TEST u.MASK] [CC=f(op1 • I)]	All 0's	All 1's	Mixed	op1=MASK	Never Set
S [SHIFT]	All 0's	op1<0	op1>0	1 Shifted From Bit 0	Never Set
I [I/O]	Never Set	Never Set	Always Set	Read Parity Error	Excep'n Rcvd

Notes:
1. One and only one of CC0, CC1 and CC2 is always set.
2. A MASK of B'111x' results in an unconditional branch. A MASK of B'0000' does not guarantee a "NO–BRANCH".
3. CC0 for ARITHMETIC–with–CARRY operations can not be SET but can be RESET.
4. A BCR or JC with a MASK of B'0000' will branch if the PSC CARRY bit is on.
5. CC3 and CCcry can be set by a COMPARE since COMPARE is a SUBTRACT without storage of results [op1–op2].
6. The CC's for CTLZ are based on the contents of R2 after the CTLZ. Thus CC1 can not be set.

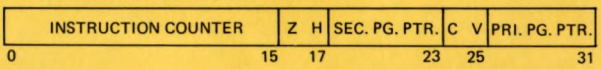
MACHINE CHECK/PROGRAM CHECK REGISTER SETTINGS

BIT 0 – I/O READ PARITY ERROR
BIT 1 – I/O DEVICE RESPONSE FAILURE
BIT 2 – STORAGE READ PARITY ERROR
BIT 3 – INVALID OP or ADDRESS
BIT 4 – ERROR DURING UCS
BIT 5 – 0, NOT USED
BIT 6 – ERROR DURING IFETCH
BIT 7 – 0

ASSEMBLY REGISTER SPECIFICATION vs MACHINE CODE TABLE

P. Page Asm No.	P. Page Mach Code	S. Page Asm No.	S. Page Mach Code	Primary Page Asm No.	Primary Page Mach Code	Primary Page Asm No.	Primary Page Mach Code	Secondary Page Asm No.	Secondary Page Mach Code	Secondary Page Asm No.	Secondary Page Mach Code
0	X'0'	16	X'1'	0	X'0'	1	X'1'	16	X'0'	17	X'1'
2	X'2'	18	X'3'	2	X'2'	3	X'3'	18	X'2'	19	X'3'
4	X'4'	20	X'5'	4	X'4'	5	X'5'	20	X'4'	21	X'5'
6	X'6'	22	X'7'	6	X'6'	7	X'7'	22	X'6'	23	X'7'
8	X'8'	24	X'9'	8	X'8'	9	X'9'	24	X'8'	25	X'9'
10	X'A'	26	X'B'	10	X'A'	11	X'B'	26	X'A'	27	X'B'
12	X'C'	28	X'D'	12	X'C'	13	X'D'	28	X'C'	29	X'D'
14	X'E'	30	X'F'	14	X'E'	15	X'F'	30	X'E'	31	X'F'
HALFWORD REGISTERS				*BYTE REGISTERS*							

PROGRAM STATUS WORD (PSW) FORMAT



PSW SAVE AREA IN GENERAL REGISTERS

PG.	REG.	ADDR.	
0	0	X'0000'	LEVEL 0 PSW SAVE AREA – I.C.
0	2	X'0002'	Z H SEC. PG. PTR. C V PRI. PG. PTR.
			0 · 2 · · 8 · 10 · · 15
1	12	X'001C'	LEVEL 7 PSW SAVE AREA – I.C.
1	14	X'001E'	Z H SEC. PG. PTR. C V PRI. PG. PTR.

PSW SWAP TIME 6.4 us

Setting of ZHCV bits of the PSW

	Z	H	C	V
arith w/o carry	$\overline{A0} \cdot \overline{A1} \cdots \overline{A7}$	$(A0 \vee V) \cdot \overline{Z}$	C0	C0 v C1
arith w/ carry	$\overline{A0} \cdot \overline{A1} \cdots \overline{A7} \cdot Z$	$(A0 \vee V) \cdot \overline{Z}$	C0	C0 v C1
Logical	$\overline{A0} \cdots \overline{A7}$	A0 ··· A7	0	0
Test	$\overline{A0} \cdots \overline{A7}$ after (R1) · M	A0 ··· A7 after (R1) v $\overline{M}$	0	$\overline{A0} \cdots \overline{A7}$ after (R1) v M
Shift	$\overline{A0} \cdots \overline{A7}$	A0	0	1 moved from bit 0
I/O	0	0	EXCEPTION Received	I/O Parity

CC0=Z CC1=H · $\overline{Z}$ CC2=$\overline{H}$ · $\overline{Z}$ CC3=V

Meaning of condition code setting and masks to use

	CC0	CC1	CC2	CC3	
MASK	1XXX	X1XX	XX1X	XXX1	0000
Arithmetic	result=0	<0	>0	overflow	carry
Compare	op1=op2	op1<op2	op1>op2	overflow	carry
Logical	all 0's	all 1's	mixed	no branch	no branch
Test	all 0's	all 1's	mixed	identical	no branch
Shift	all 0's	hi order 1	positive	1 moved from bit 0	no branch
I/O	no branch	no branch	branch	I/O parity	EXCEPTION Received