UC Reference Summary

Release 110

IBM CONFIDENTIAL

10/1/76

SUMMARY OF UC PROGRAMMING SUPPORT TOOLS FOR VM/CMS ENVIRONMENTS

UC ASSEMBLER

UCASM fname [options]

options:

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

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

options:

ALOGIC / NOALOGIC	RLD / NORLD	NUM / NONUM
ESD / NOESD	XREF / NOXREF / XREF SHORT	STMT / NOSTMT
LIST / NOLIST	DECK / NODECK	TERM / NOTERM
MCALL / NOMCALL	OBJ / NOOBJ	ALIGN / NOALIGN
MLOGIC / NOMLOGIC	TEST / NOTEST	CONF / NOCONF
LIBMAC / NOLIBMAC		

[FLAG nnn]

[LINECNT nn]

SUBSET UC1 / BASIC / BYTE / THEBES

[SYSPARM (...)]

[DISKUT [Y] [1] [2] [3]]

UC LINKAGE EDITOR

UCLINK fname [options]

options:

SYM / NOSYM	[CLEAR0]
	[EPOINT]
ESD / NOESD	[COL71]
	[ID]
TYPE / NOTYPE	[DOCUMENT]
	[Knnn]
XREF / NOXREF	[NOCOM]

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 textfn1 . . . textfnx

TEXTDICT ADD $dictfn textfn1 . . . textfnx

TEXTDICT DEL $dictfn textfn1 . . . textfnx

TEXTDICT LIST $dictfn

UC PROGRAM DEBUG SIMULATOR

INVOCATION – UCPDS

UCPDS Sentence Formats

Comment Sentence

* comment line

Immediate Sentence

command1;command2; . . . commandn

Conditional Sentence

label: 'ON' command;command1; . . . commandn

Immediate commands will be executed when the 'ON' command conditions are true.

UCPDS Commands

- UC modelname mainstore-size regspace-size [uc idname]

  modelname: UC-0 mainstore-size: 1-512K or 1-524288 decimal
  UC.5
  UC.5E
  UC.5M regspace-size: 1-4K or 1-4096 decimal
  UC-1
  THEBES

  uc idname: 1 to 8 alphabetic characters

- ADAPTER device type interrupt level address [device specific parameters]

- LOAD membername[,membername, . . .]

- SET facilityname
  memory address value
  register address
  segment address

- DISPLAY facilityname[, . . .]
  memory address
  register address
  segment address

- RETRIEVE [L] [O] membername [,membername . . .]

- GO

- PAUSE (conditional sentences only)

- TRACE (conditional sentences only)

- MAP (conditional sentences only)
  MAPD
  MAPC

- WAIT

- IOINT ±n

- CLEAR { label / *ALL } [,label . . .]

- INCREMENT { CTR1 : CTR5 } [, { CTR1 : CTR5 }]
  DECREMENT

- IOACK

- END

- CHKPOINT device id (CMS only)

- RESTART [device id] (CMS only if device id specified)

- DO label [,label . . .]

- **ON** event [(test expression[,test expression])]
  [,event[(test expression . . .)]]

event:

ISTEP	(default)	BR
SWAP		MC
TA		PC
TD		CSG
WAIT		EWAIT
		HANG

test expression format:

facilityname / memory address / register address

= EQ, ¬ = NE } value[:value]

> GT, ¬ > NG, > = GE, < LT, ¬ < NL, < = LE } value

Facility Names & Usage

CLOCK	S	D	
CLR		D	T
CMNMSK	S	D	
CTR1-5	S	D	T
IC	S	D	T
INST		D	
IOBUS	S	D	
IOIRR	S	D	
LABELS		D	
LEVELS		D	T
LLR		D	T
MCPC	S	D	T
MMB	S	D	
OLNCNT	S	D	
OPCODE		D	
PIC		D	T
PIRR	S	D	
PRYPGE	S	D	
PSW0-7	S	D	
ROS	S	D	
ROSF	S		
SDYPGE	S	D	
TIMEOUT	S	D	
UCID	S	D	
ZHCV		D	T

UC1 Only Facilities

DAL	S	D	T
EPSW0-7	S	D	
EPSWS0-7	S	D	
IAL	S	D	T
MCF	S	D	T
MCF0-7	S	D	
MCFS0-7	S	D	
PCR	S	D	T
PSWS0-7	S	D	

Memory Address Formats & Usage

M (hexaddr)	S	D	T
M (&label+hexdispl)	S	D	T
M (&csect*label+hexdispl)	S	D	T
M (&csect*label+hexdispl [:memory address])		D	
M (hexaddr:hexaddr)		D	
M (hexaddr (decimal length))		D	

Register Address Formats & Usage

R (ppr)	S	D	T
RE (ppr)	S	D	T
RP (pp)	S	D	
RPE (pp)	S	D	
RPP	S	D	
RPEP	S	D	
RPS	S	D	
RPES	S	D	
R (ppr:ppr)		D	
RE (ppr:ppr)		D	
R (ppr(dec.len.))		D	
RE (ppr(dec.len.))		D	

Segment Address Format & Usage

S (hexaddr)	S	D	
S (hexaddr: hexaddr)		D	
S (hexaddr (decimal length))		D	

Usage: S = Settable
D = Displayable
T = Testable

Notes: ppr = page & register numbers
pp = page number

UC MACHINE INSTRUCTIONS

UC-0, UC.5, THEBES & UC1

	Instruction	Mnemonic	Operands	Op+Extn	CC	PC
X	Add Register Immediate	ARI	P,I8	3	A	
	Add Register	AR	PSP,PSP	7 8,9x	A	
	Add with Carry Register	AYR	PSP,PSP	7 A,Bx	A	
	And Register Immediate	NRI	P,I8	0	L	
	And Register	NR	PSP,PSP	7 0,1x	L	
X	Branch on Condition Register	BCR	M4,AH	A 0		
	Branch and Link Register	BALR	H,AH	A 3		
	Compare Register	CR	PSP,PSP	7 6,7x	C	
	Control Immediate	KI	PE,I8	6		PR
	Exclusive Or Reg Immediate	XRI	P,I8	2	L	
	Exclusive Or Register	XR	PSP,PSP	7 4,5x	L	
	I/O Immediate	IOI	PO,I8	6	I	I
	I/O	IO	P,H	A 4	I	I
X	Jump on Condition	JC	M4,S	9 0y		
	Load Register Immediate	LRI	P,I8	4		
	Load Indirect	LN	P,AH	A B		
	Load Register Space Indirect	LRN	P,H	A F		PR
	Load Register	LR	RA,RA	8 4–7x		
	Or Register Immediate	ORI	P,I8	1	L	
	Or Register	OR	PSP,PSP	7 2,3x	L	
	Rotate Left	RL	RA,I3	8 2,3x	S	
	Shift Left Logical	SLL	RA,I3	8 0,1x	S	
	Store Indirect	STN	P,AH	A 8		
	Store Register Space Indirect	STRN	P,H	A C		PR
	Subtract Register	SR	PSP,PSP	7 C,Dx	A	
	Subtract with Carry Register	SYR	PSP,PSP	7 E,Fx	A	
	Test Register Immediate	TRI	P,I8	5	T	

UC.5, THEBES & UC1

Instruction	Mnemonic	Operands	Op+Extn	CC	PC
Add Halfword Reg Immediate	AHRI	H,I4	C D	A	
Add Halfword Register	AHR	H,H	C 8	A	
Add with Carry Halfword Reg	AYHR	H,H	C A	A	
And Halfword Register	NHR	H,H	C 0	L	
Branch and Count Register	BCTR	P,AH	A 2		
Compare Halfword Register	CHR	H,H	C 6	C	
Count Leading Zeros	CTLZ	H,H	C 1	L	
Exclusive Or Halfword Register	XHR	H,H	C 4	L	
I/O Halfword	IOH	H,H	A 5	I	I
Jump on Bit Zero	JBZ	BIT,S	9 1y		
Load Halfword Indirect	LHN	H,AH	A 9		
Load Halfword Short	LHS	H,AS	B 1y		
Load Hwd Reg Space Indirect	LHRN	H,H	A D		PR
Load Halfword Register	LHR	H,H	C 3		
Or Halfword Register	OHR	H,H	C 2	L	
Rotate Left Halfword	RLH	H,I4	C B	S	
Shift Left Halfword Logical	SLHL	H,I4	C 9	S	
Store Halfword Indirect	STHN	H,AH	A A		
Store Halfword Short	STHS	H,AS	B 0y		
Store Hwd Reg Space Indirect	STHRN	H,H	A E		PR
Test and Set	TS	0,AH	A 1	L	
Subtract Halfword Register	SHR	H,H	C C	A	
Subtract with Carry Hwd Reg	SYHR	H,H	C E	A	
Subtract Halfword Reg Immediate	SHRI	H,I4	C F	A	

X = Extended mnemonics available for this instruction

- ON event [(test expression[,test expression])]
  [,event[(test expression . . .)]]

  event: ISTEP (default), SWAP, TA, TD, WAIT, BR, MC, PC, CSG, EWAIT, HANG

  test expression format:

  facilityname / memory address / register address
  = EQ, ¬ = NE } value[:value]
  > GT, ¬ > NG, > = GE, < LT, ¬ < NL, < = LE } value

Facility Names & Usage

CLOCK	S	D	
CLR		D	T
CMNMSK	S	D	
CTR1-5	S	D	T
IC	S	D	T
INST		D	
IOBUS	S	D	
IOIRR	S	D	
LABELS		D	
LEVELS		D	T
LLR		D	T
MCPC	S	D	T
MMB	S	D	
OLNCNT	S	D	
OPCODE		D	
PIC		D	T
PIRR	S	D	
PRYPGE	S	D	
PSW0-7	S	D	
ROS	S	D	
ROSF	S		
SDYPGE	S	D	
TIMEOUT	S	D	
UCID	S	D	
ZHCV		D	T

UC1 Only Facilities

DAL	S	D	T
EPSW0-7	S	D	
EPSWS0-7	S	D	
IAL	S	D	T
MCF	S	D	T
MCF0-7	S	D	
MCFS0-7	S	D	
PCR	S	D	T
PSWS0-7	S	D	

Memory Address Formats & Usage

M (hexaddr)	S	D	T
M (&label+hexdispl)	S	D	T
M (&csect*label+hexdispl)	S	D	T
M (&csect*label+hexdispl [:memory address])		D	
M (hexaddr:hexaddr)		D	
M (hexaddr (decimal length))		D	

Register Address Formats & Usage

R (ppr)	S	D	T
RE (ppr)	S	D	T
RP (pp)	S	D	
RPE (pp)	S	D	
RPP	S	D	
RPEP	S	D	
RPS	S	D	
RPES	S	D	
R (ppr:ppr)		D	
RE (ppr:ppr)		D	
R (ppr(dec.len.))		D	
RE (ppr(dec.len.))		D	

Segment Address Format & Usage

S (hexaddr)	S	D	
S (hexaddr: hexaddr)		D	
S (hexaddr (decimal length))		D	

Usage: S = Settable
D = Displayable
T = Testable

Notes: ppr = page & register numbers
pp = page number

UC MACHINE INSTRUCTIONS

UC-0, UC.5, THEBES & UC1

	Instruction	Mnemonic	Operands	Op+Extn	CC	PC
X	Add Register Immediate	ARI	P,I8	3	A	
	Add Register	AR	PSP,PSP	7 8,9x	A	
	Add with Carry Register	AYR	PSP,PSP	7 A,Bx	A	
	And Register Immediate	NRI	P,I8	0	L	
	And Register	NR	PSP,PSP	7 0,1x	L	
X	Branch on Condition Register	BCR	M4,AH	A 0		
	Branch and Link Register	BALR	H,AH	A 3		
	Compare Register	CR	PSP,PSP	7 6,7x	C	
	Control Immediate	KI	PE,I8	6		PR
	Exclusive Or Reg Immediate	XRI	P,I8	2	L	
	Exclusive Or Register	XR	PSP,PSP	7 4,5x	L	
	I/O Immediate	IOI	PO,I8	6	I	I
	I/O	IO	P,H	A 4	I	I
X	Jump on Condition	JC	M4,S	9 0y		
	Load Register Immediate	LRI	P,I8	4		
	Load Indirect	LN	P,AH	A B		
	Load Register Space Indirect	LRN	P,H	A F		PR
	Load Register	LR	RA,RA	8 4–7x		
	Or Register Immediate	ORI	P,I8	1	L	
	Or Register	OR	PSP,PSP	7 2,3x	L	
	Rotate Left	RL	RA,I3	8 2,3x	S	
	Shift Left Logical	SLL	RA,I3	8 0,1x	S	
	Store Indirect	STN	P,AH	A 8		
	Store Register Space Indirect	STRN	P,H	A C		PR
	Subtract Register	SR	PSP,PSP	7 C,Dx	A	
	Subtract with Carry Register	SYR	PSP,PSP	7 E,Fx	A	
	Test Register Immediate	TRI	P,I8	5	T	

UC.5, THEBES & UC1

Instruction	Mnemonic	Operands	Op+Extn	CC	PC
Add Halfword Reg Immediate	AHRI	H,I4	C D	A	
Add Halfword Register	AHR	H,H	C 8	A	
Add with Carry Halfword Reg	AYHR	H,H	C A	A	
And Halfword Register	NHR	H,H	C 0	L	
Branch and Count Register	BCTR	P,AH	A 2		
Compare Halfword Register	CHR	H,H	C 6	C	
Count Leading Zeros	CTLZ	H,H	C 1	L	
Exclusive Or Halfword Register	XHR	H,H	C 4	L	
I/O Halfword	IOH	H,H	A 5	I	I
Jump on Bit Zero	JBZ	BIT,S	9 1y		
Load Halfword Indirect	LHN	H,AH	A 9		
Load Halfword Short	LHS	H,AS	B 1y		
Load Hwd Reg Space Indirect	LHRN	H,H	A D		PR
Load Halfword Register	LHR	H,H	C 3		
Or Halfword Register	OHR	H,H	C 2	L	
Rotate Left Halfword	RLH	H,I4	C B	S	
Shift Left Halfword Logical	SLHL	H,I4	C 9	S	
Store Halfword Indirect	STHN	H,AH	A A		
Store Halfword Short	STHS	H,AS	B 0y		
Store Hwd Reg Space Indirect	STHRN	H,H	A E		PR
Test and Set	TS	0,AH	A 1	L	
Subtract Halfword Register	SHR	H,H	C C	A	
Subtract with Carry Hwd Reg	SYHR	H,H	C E	A	
Subtract Halfword Reg Immediate	SHRI	H,I4	C F	A	

X = Extended mnemonics available for this instruction

THEBES & UC1

Instruction	Mnemonic	Operands	Op+Extn	CC	PC
Branch Indirect Indexed	BNX	P,AH	C 5		
Load Byte with Decrement	LND	P,AH	8 F		
Load Byte with Increment	LNI	P,AH	8 B		
Load Halfword with Decrement	LHND	H,AH	8 D		
Load Halfword with Increment	LHNI	H,AH	8 9		
Store Byte with Decrement	STND	P,AH	8 C		
Store Byte with Increment	STNI	P,AH	8 8		
Store Halfword with Decrement	STHND	H,AH	8 E		
Store Halfword with Increment	STHNI	H,AH	8 A		

UC1

Instruction	Mnemonic	Operands	Op+Extn	CC	PC
Add w/Carry Hwd Reg Extn	AYHRE	HE,HE	F 9	A	
Branch and Link	BAL	AH,AS	D 7		
Branch on Condition	BC	M4,AS	D 6		
Compare w/Carry Hwd Reg Extn	CYHRE	HE,HE	F 8	C	
Compare Logical Byte Storage	CLS	AH+,AH+,H	E 8	U	
Compare Logical Hwd Storage	CLHS	AH+,AH+,H	E 8	U	
Control Direct Out	KDO	I4	F F		
Divide Halfword Register	DHR	HP,H	E 0		
Jump and Link	JAL	AH,S	D 7		
Jump on Condition Extended	JCX	M4,S	D 6		
Load Address	LA	W,AS	D B		
Load Byte	L	P,AS	D 0		
Load Halfword	LH	H,AS	D 2		
Load Hwd Register Quadrant	LHQ	Q,AH	8 3		
Load Word	LW	W,AS	D 4		
Load Halfword Register Extn	LHRE	HE,H	F B		
Load Virtual Access Code and Translate Entry	LAT	W,H+	E 1		PR
Move Byte Storage	MVS	AH+,AH+,H	E 8		
Move Halfword Storage	MVHS	AH+,AH+,H	E 8		
Move Halfword Register Extn	MVHRE	HE,HE	F D		
Multiply Halfword Register	MHR	H,H	E 0		
Store Byte	ST	P,AS	D 1		
Store Halfword	STH	H,AS	D 3		
Store Halfword Reg Quadrant	STHQ	Q,AH	8 2		
Store Word	STW	W,AS	D 5		
Store Halfword Register Extn	STHRE	HE,H	F C		
Store Virtual Access Code and Translate Entry	STAT	W,H+	E 1		PR
Subtract w/Carry Hwd Reg Extn	SYHRE	HE,HE	F A	A	

OPERAND KEY

AS — Either SA or Displacement (Register) where D (R) can be
- DH5 (BS) – LHS and STHS
- D16 (AH) – BC, BAL, L, LH, ST, STH, and STW
- D16 (W) – LA and LW

AH — Halfword register in the primary or secondary page which is used as an address register

BS — Base register specification 12, 14, 28, 30

BIT — Bit to be tested in halfword register 2 (value 0 to 15)

Dnn — Displacement value in bytes, where nn specifies the number of bits available

DHnn — Displacement value in halfwords, where nn specifies the number of bits available

H — Halfword register, value 0 to 30 (even)

HE — Halfword register extension

HP — The first of two contiguous halfword register, where HP begins on a fullword boundary

Inn — Immediate field of bit length nn

Mn — Mask field of bit length n

P — Primary page byte register, value 0 to 15

PE — Even primary page byte register, value 0 to 14 (even)

PO — Odd primary page byte register, value 1 to 15 (odd)

PSP — PSP,PSP indicates byte registers in either the primary or secondary page, but both registers must be in the same page

RA — Any byte register, 0 to 31

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 next sequential instruction should be in the range:
- JC and JBZ – -128 to +126
- JAL and JCX – -32768 to +32767 (halfwords)

SA — (Storage Address) Relocatable of absolute expression. SA minus the value of the base register contents should be in the range of the allowable displacement for the instruction in question.
Displacement field lengths:
- DH5 – LHS and STHS
- D16 – LA, L, LH, LW, STH, STW

W — Fullword register formed by concatenating the halfword register specified by W to its register extension

x — Bit 15 of the instruction is used to indicate that the register page used is
0 – primary 1 – secondary

y — Bit 15 is used to distinguish operations

Q — Specifies one of the four active halfword register quadrants
- 0 – Primary page, basic halfword registers
- 1 – Secondary page, basic halfword registers
- 2 – Primary page, extension halfword registers
- 3 – Secondary page, extension halfword registers

CC = CONDITION CODE SETTINGS

- A – Arithmetic
- L – Logical
- I – I/O
- S – Shift
- T – Test under Mask
- U – Logical (Unsigned) Compare
- C – Algebraic Compare

PC = Privilege Code

- PR – Instructions affected by the UC1 process mode bits of the PSW
- I – Programs ability to execute PIO and EPIO instructions as well as, KI 35 and 37 is affected by the UC1 process mode bits of the PSW

UNIVERSAL CONTROLLER INSTRUCTION FORMATS

	0 3	4 5 7	8 11	12 14 15
NRI	0	P	I8	
ORI	1	P	I8	
XRI	2	P	I8	
ARI	3	P	I8	
LRI	4	P	I8	
TRI	5	P	I8	
IOI	6	PO	I8	
KI	6	PE	I8	
NR	7	PSP	PSP	0/1 (PP/SP)
OR	7	PSP	PSP	2/3
XR	7	PSP	PSP	4/5
CR	7	PSP	PSP	6/7
AR	7	PSP	PSP	8/9
AYR	7	PSP	PSP	A/B
SR	7	PSP	PSP	C/D
SYR	7	PSP	PSP	E/F
SLL	8	0 I3	RA (PP)	0
SLL	8	0 I3	RA (SP)	1
RL	8	0 I3	RA (PP)	2
RL	8	0 I3	RA (SP)	3
STHQ	8	1 Q	AH	2
LHQ	8	1 Q	AH	3
LR	8	RA (PP)	RA (PP)	4
LR	8	RA (PP)	RA (SP)	5
LR	8	RA (SP)	RA (PP)	6
LR	8	RA (SP)	RA (SP)	7
STNI	8	P	AH	8
LHNI	8	H	AH	9
STHNI	8	H	AH	A
LNI	8	P	AH	B
STND	8	P	AH	C
LHND	8	H	AH	D
STHND	8	H	AH	E
LND	8	P	AH	F

	0 3	4 5 7	8 11	12 14	15
JC	9	M4	I8		0
JBZ	9	BIT	I8		1
BCR	A	M4	AH	0	
TS	A	0	AH	1	
BCTR	A	P	AH	2	
BALR	A	H	AH	3	
IO	A	P	H	4	
IOH	A	H	H	5	
STN	A	P	AH	8	
LHN	A	H	AH	9	
STHN	A	H	AH	A	
LN	A	P	AH	B	
STRN	A	P	H	C	
LHRN	A	H	H	D	
STHRN	A	H	H	E	
LRN	A	P	H	F	
STHS	B	H	AH	DH5	0
LHS	B	H	AH	DH5	1
NHR	C	H	H	0	
CTLZ	C	H	H	1	
OHR	C	H	H	2	
LHR	C	H	H	3	
XHR	C	H	H	4	
BNX	C	H	AH	5	
CHR	C	H	H	6	
AHR	C	H	H	8	
SLHL	C	I4	H	9	
AYHR	C	H	H	A	
RLH	C	I4	H	B	
SHR	C	H	H	C	
AHRI	C	I4	H	D	
SYHR	C	H	H	E	
SHRI	C	I4	H	F	

	0 4	8 12	16 18 20 22 24 28 31
L	D P	AH 0	D16
ST	D P	AH 1	D16
LH	D H	AH 2	D16
STH	D H	AH 3	D16
LW	D W	AH 4	D16
STW	D W	AH 5	D16
JCX	D M4	0 6	D15 Sign
BC	D M4	AH 6	D15 Sign
BAL	D AH	AH 7	D15 Sign
JAL	D AH	0 7	D15 Sign
LA	D W	W B	D16
MHR	E H	H 0	2 3 3 0 01
DHR	E HP	H 0	2 3 3 0 03
LAT	E W	H 1	2 3 1 0 0C
STAT	E W	H 1	2 3 1 0 0D
MVS	E AH	AH 8	1 1 1 0 H 9
MVHS	E AH	AH 8	2 1 1 0 H 9
CLS	E AH	AH 8	1 1 1 0 H F
CLHS	E AH	AH 8	2 1 1 0 H F
CYHRE	F HE	HE 8	
AYHRE	F HE	HE 9	
SYHRE	F HE	HE A	
LHRE	F HE	H B	
STHRE	F HE	H C	
MVHRE	F HE	HE D	
KDO	F I4	0 F	
PC	F F	F F	

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 + (1.2n) / 2.4 (n=0)
RLH	1.6 + (1.2n) / 2.4 (n=0)
SLL	1.6 + (1.2n) / 2.4 (n=0)
SLHL	1.6 + (1.2n) / 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 + 1.6t
CS write	6.4 + 1.6t

n = 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

** K1 – Read/Write Page Pointers	2.8/2.8
K1 – 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 for Subsets, Thebes, Byte, and Basic — LRI &SYSSCRT.K(S)
LRI &SYSSCRT+1,L(S)
BCR MASK,&SYSSCRT
Load address into e/o pair of registers

and for remaining subsets to — JCX MASK,S

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

ASSIGNED CONTROL ADDRESSES

NO.	WRITE OPERATIONS		NO.	READ OPERATIONS	
0	Reset Master Mask		1	Read Master Mask	(T)
2	Write Common Mask, Basic		3	Read Common Mask, Basic	
4	OR to PIRR, Basic		5	Read PIRR, Basic	
6	AND to PIRR, Basic		7	Read IOIRR, Basic	
8	Write MC/PC Status		9	Read MC/PC Status	
10	Write Primary Page Number		11	Read Primary Page Number	
12	Write Secondary Page Number		13	Read Secondary Page Number	
14	Set Master Mask		15	Read Levels	
24	Reset Integ Chnl Mask	(T, 1)	25	Read Integ Chnl Mask	(T, 1)
26	Write PSC	(5, T, 1)	27	Read PSC	(5, T, 1)
28	Write Next Level	(5, T, 1)	39	Swap Page Pointers	(T)
30	Adapter Reset	(T)	53	Execute Halfword Register	(T)
32	System Reset	(T)	121	Read PCR, Basic	(1)
35	Set Current Lvl in PIRR	(T, 1)	127	Swap PSW	(1)
37	Reset Current Lvl in PIRR	(T, 1)			
38	Set Integ Chnl Mask	(T, 1)			

All control addresses listed above apply to all engines from UC–0 upwards unless indicated otherwise by a parenthesized engine specification. Eg., the notation (5, T, 1) indicates that this control address is available on only the UC.5, Thebes and UC1 engines.

Contents of the first operand register (where meaningful).

Interrupt Sources:

BIT	PIRR & IOIRR	MC/PC
0	Int. level 0	I/O
1	Int. level 1	External
2	Int. level 2	Storage
3	Int. level 3	Process
4	Int. level 4	Integrated Channel
5	Int. level 5	Internal
6	Int. level 6	Instruction Counter Code
7	Int. level 7	Reserved

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)
J	Address	Byte 1 of a four byte address (bits 8–15)

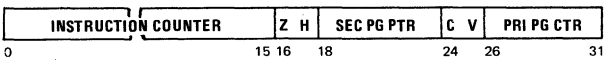
DEFINING CONSTANTS

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
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
J	Address	Byte 1 of a four byte address (bits 8–15)
W	Address	Bytes 0 and 1 of a four byte address (bits 0–15)

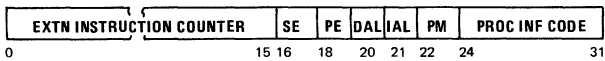
ASSEMBLY REGISTER SPECIFICATION vs MACHINE CODE TABLE

HALF REGISTERS				*BYTE REGISTERS*							
P. Page		S. Page		Primary Page				Secondary Page			
Asm No.	Mech Code	Asm No.	Mech Code	Asm No.	Mech Code	Asm No.	Mech Code	Asm No.	Mech Code	Asm No.	Mech 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'

PROCESS STATUS WORD (PSW) FORMAT



EXTENDED PROCESS STATUS WORD (EPSW) FORMAT



SE = Secondary Page Pointer Extension (high 2 bits)
PE = Primary Page Pointer Extension (high 2 bits)
DAL = Data Address Length
IAL = Instruction Address Length
PM = Process Mode Field
- 00 Control Mode with memory access privilege
- 01 Control Mode
- 10 Application Mode
- 11 I/O Mode

CONDITION CODE MEANINGS

PSC	Z	$\overline{Z} \bullet H$	$\overline{Z} \bullet \overline{H}$	V	C
Cond Code	CC0	CC1	CC2	CC3	
Branch Mask	1XXX	X1XX	XX1X	XXX1	0000
Arithmetic	Result Zero	Result Negative	Result Positive	Overflow	Carry
Algebraic Compare	Equal	Less Than	Greater	Overflow	Carry
Logical Compare	Equal	Less Than	Greater	Reset	Carry
Logical	Result All Zeros	Result All Ones	Result Mixed	Reset	Reset
Test Under Mask	All Zeros Under Mask	All Ones Under Mask	Mixed Under Mask	Identical to Mask	Reset
Shift	Result All Zeros	Result Has High Order One	Result Mixed with High Order Zero	A One Bit was Shifted Out of the Register	Reset
Input/ Output	Reset	Reset	Always Set	Read Parity Error	Exception Received

Notes:

1. One and only one of the conditions CC0-CC2 will always exist. CC3 is independent and may coexist with these.
2. A mask of '111X' results in an unconditional branch. A mask of '0000' tests the condition of the Carry Latch.
3. Arithmetic condition code 0 (CC0) is cumulative for the "with carry" operations. A "with carry" operation cannot initiate a CC0, though it may continue it.
4. Arithmetic condition codes CC0-CC2 (zero, negative, and positive) will indicate the true sign of the result, even if there is an overflow, with the following single exception: the addition of two maximum negative values will set CC0 instead of CC1. This result is normal, and is required to produce correct results when doing multiple instruction arithmetic.
5. Some engine models do not set the condition code on I/O operations.

LOGICAL SETTINGS OF THE PROCESS STATUS CODE (PSC)

Instruction Type	Z	H	C	V
Arithmetic "without carry" including Compare	$\overline{A0} \bullet \overline{A1} \bullet \cdots \bullet \overline{A7}$	A0 ¥ V	C0	C0 ¥ C1
Arithmetic "with carry" including Compare	$\overline{A0} \bullet \overline{A1} \bullet \cdots \bullet \overline{A7} \bullet Z'$	A0 ¥ V	C0	C0 ¥ C1
Logical Compare	$\overline{A0} \bullet \overline{A1} \bullet \cdots \bullet \overline{A7}$	$\overline{C0}$	C0	Zero
Logical	$\overline{A0} \bullet \overline{A1} \bullet \cdots \bullet \overline{A7}$	$A0 \bullet A1 \bullet \cdots \bullet A7$	Zero	Zero
Test Under Mask	$\overline{A0} \bullet \overline{A1} \bullet \cdots \bullet \overline{A7}$ following Byte • Mask	$A0 \bullet A1 \bullet \cdots \bullet A7$ following Byte V $\overline{MASK}$	Zero Zero	$\overline{A0} \bullet \overline{A1} \bullet \cdots \bullet \overline{A7}$ following Byte ¥ MASK
Shift or Rotate	$\overline{A0} \bullet \overline{A1} \bullet \cdots \bullet \overline{A7}$	A0	Zero	A one (1) bit was shifted out of the register
I/O	Zero	Zero	Exception Received	Read Parity Error

Notes:

1. Z' means the previous value of the Z bit.
   Ci means the carry out of the ith bit position.
   Ai means the ith bit of the answer.

   For byte operations i ranges from 0 to 7 with bit 0 being the leftmost bit and bit 7 being the rightmost or least significant bit. For halfword operands i ranges from 0 to 15 with bit 0 being most significant and bit 15 being least significant.
2. The above definitions are for use with byte operands. Definitions for use with halfword operands are the same except substitute A15 wherever A7 appears.
3. The symbols for the PSC bits are taken from the following usual meanings for the corresponding bits: — Zero, High bit, Carry, Overflow.

Notes

Notes