

OPC	MNE	INSTRUCTION	TYPE	DESCRIPTION
0020AB	ACR	ADD CONSTANT REGISTER	CR	$A1 \leftarrow (A1) + (\text{CONST}(A, B))$
002070	AIR	ADD INDEX REGISTER	CR	$A1 \leftarrow (A1) + (\text{AER}, \text{IAR})$
0021AB	LCR	LOAD CONSTANT REGISTER	CR	$A1 \leftarrow (\text{CONST}(A, B))$
002170	RIR	READ INDEX REGISTER	CR	$A1 \leftarrow (\text{AER}, \text{IAR})$
002270	IIR	INCREMENT INDEX REGISTER	CR	$\text{IAR} \leftarrow \text{AER}, \text{IAR} + (A1); A1 \leftarrow \text{IAR} + (A1); \text{AER} \leftarrow 0$
002277	TIR	TRANSFER INDIRECT ADR REG	CR	$\text{IAR} \leftarrow A[0:7]; \text{AER} \leftarrow 0$
0024AB	SCR	SUBTRACT CONSTANT REGISTER	CR	$A1 \leftarrow A1 - (\text{CONST}(A, B))$
002470	SIR	SUBTRACT INDIRECT REGISTER	CR	$A1 \leftarrow (A1) - (\text{AER}, \text{IAR})$
0025AB	LNR	LOAD NEGATIVE REGISTER	CR	$A1 \leftarrow -(\text{CONST}(A, B))$
0030CC	ADC	ADD CHARACTER	AR	$A1[0:23] \leftarrow A1[0:23] + \text{CC}$
003001	IAC	INCREMENT A1	CR	$A1 \leftarrow (A1) + 1$
003020	ANM	ADD NUMERIC MASK	CR	$A1 \leftarrow (A1) + 20$
0031CC	LDC	LOAD CHARACTER	AR	$A1 \leftarrow 0; A1[0:5] \leftarrow \text{CC}$
003101	CLI	CLEAR AND INCREMENT	CR	$A1 \leftarrow 1$
0032CC	TAC	TRANSFER CHARACTER TO IAR	CH	$A1 \leftarrow \text{CC}; \text{AER}, \text{IAR} \leftarrow \text{CC}$
0033CC	ZIR	ZERO IAR	CH	$A1 \leftarrow 0; \text{AER}, \text{IAR} \leftarrow 0$
0034CC	SBC	SUBTRACT CHARACTER	AR	$A1[0:23] \leftarrow A1[0:23] - \text{CC}$
003401	DAC	DECREMENT A1	CR	$A1 \leftarrow (A1) - 1$
003420	RNM	REMOVE NUMERIC MASK	CR	$A1 \leftarrow (A1) - 20$
0035CC	LNC	LOAD NEGATIVE CHARACTER	AR	$A1 \leftarrow 0; A1[0:23] \leftarrow -\text{CC}$
003501	CLD	CLEAR AND DECREMENT	CR	$A1 \leftarrow -1$
004CCC	STA	STORE A1	CA	$\text{MEM}(\text{ADDR}) \leftarrow (A1)$
006CCC	STX	STORE X REGISTER	CA	$\text{MEM}(\text{ADDR}) \leftarrow (X)$
010000	TTY	OPERATE TELETYPE	TY	OPERATE TELETYPE
010001	KRB	KEYBOARD READ BUFFER	TY	$A1[0:5] \leftarrow (\text{IB}[0:5]) \vee (A1)$
010002	KCB	KEYBOARD CLEAR BUFFER	TY	$\text{IB}[0:5] \leftarrow 0$
010004	KCF	KEYBOARD CLEAR FLAG	TY	$\text{IF} \leftarrow 0$
010007	KRD	KEYBOARD READ DYNAMIC	TY	$A1[0:5] \leftarrow (\text{IB}[0:5]) \vee (A1); \text{IE}[0:5] \leftarrow 0; \text{IF} \leftarrow 0$
010010	TLB	TELEPRINTER LOAD BUFFER	TY	$\text{OB}[0:5] \leftarrow (A1[0:5]) \vee (\text{OB}[0:5])$
010020	TPT	TELEPRINTER PRINT	TY	$\text{CNTRL} \leftarrow 1$
010030	TLP	TELEPRINTER LOAD AND PRINT	TY	$\text{OB}[0:5] \leftarrow (A1[0:5]) \vee (\text{OB}[0:5]); \text{CNTRL} \leftarrow 1$
014400	VFY	VERIFY	IO	VERIFY PAPER TAPE
014600	RWD	REWIND	IO	REWIND PAPER TAPE
015000	RPT	READ PAPER TAPE	IO	READ PAPER TAPE
016000	CTR	CORE TRANSFER	IO	TRANSFER CORE TO OR FROM BUERCUGHS
020CCC	XOR	EXCLUSIVE OR	CA	$A1 \leftarrow (A1) \text{ XOR } (\text{MEM}(\text{ADDR}))$
022CCC	LIR	LOAD INDEX REGISTER	CA	$\text{IAR} \leftarrow (\text{MEM}(\text{ADDR}))$
024CCC	JAI	JUMP ADDRESS INDIRECT	CA	$\text{PAR} \leftarrow (\text{MEM}(\text{ADDR}))$
0300AB	CCR	COMPARE CONSTANT REGISTER	CA	COMPARE A1 TO $\text{CONST}(A, B)$
0301AB	CNR	COMPARE NEGATIVE REGISTER	CA	COMPARE A1 TO $-(\text{CONST}(A, B))$
0302CC	CMC	COMPARE CHARACTER	CH	COMPARE A1 TO CC
0306CC	CNC	COMPARE NEGATIVE CHARACTER	CA	COMPARE A1 TO $-\text{CC}$
04DDDD	DRD	DRUM READ	DA	$A1 \leftarrow (\text{DRUM}(\text{ADDR}))$
060CCC	CLA	CLEAR AND ADD	CA	$A1 \leftarrow (\text{MEM}(\text{ADDR}))$
061CCC	UNL	UNITE LOGICAL	CA	$A1 \leftarrow \text{MEM}(\text{ADDR}) \vee (A1) \vee (\text{MEM}(\text{ADDR}))$
062CCC	CLS	CLEAR AND SUBTRACT	CA	$A1 \leftarrow -(\text{MEM}(\text{ADDR}))$
063CCC	NOR	NEGATIVE OR	CA	$\text{MEM}(\text{ADDR}) \leftarrow (A1) \vee (\text{MEM}(\text{ADDR})); A1 \leftarrow \sim(\text{MEM}(\text{ADDR}))$
064CCC	ADD	ADD	CA	$A1 \leftarrow (A1) + (\text{MEM}(\text{ADDR}))$
065CCC	SUB	SUBTRACT	CA	$A1 \leftarrow (A1) - (\text{MEM}(\text{ADDR}))$

060CCC	CLA	CLEAR AND ADD	CA	$A1 \leftarrow (\text{MEM}(\text{ADDR}))$
061CCC	UNL	UNITE LOGICAL	CA	$A1 \leftarrow \text{MEM}(\text{ADDR}) \leftarrow (A1) \vee (\text{MEM}(\text{ADDR}))$
062CCC	CLS	CLEAR AND SUBTRACT	CA	$A1 \leftarrow -(\text{MEM}(\text{ADDR}))$
063CCC	NOR	NEGATIVE OR	CA	$\text{MEM}(\text{ADDR}) \leftarrow (A1) \vee (\text{MEM}(\text{ADDR})); A1 \leftarrow \sim(\text{MEM}(\text{ADDR}))$
064CCC	ADD	ADD	CA	$A1 \leftarrow (A1) + (\text{MEM}(\text{ADDR}))$
066CCC	SUB	SUBTRACT	CA	$A1 \leftarrow (A1) - (\text{MEM}(\text{ADDR}))$
070CCC	CMP	COMPARE	CA	COMPARE A1 TO (MEM(ADDR))
070CCC	SKP	COMPARE SKIP	CA	COMPARE A1 TO (MEM(ADDR))
0700CC	LDX	LOAD X	CA	$X \leftarrow (\text{MEM}(\text{ADDR}))$
071CCC	SGT	SKIP IF GREATER	CA	IF (A1) > (MEM(ADDR)) THEN PAR ← (PAR) + 2
072CCC	SEQ	SKIP IF EQUAL	CA	IF (A1) = (MEM(ADDR)) THEN PAR ← (PAR) + 2
073CCC	SGE	SKIP IF GREATER OR EQUAL	CA	IF (A1) ≥ (MEM(ADDR)) THEN PAR ← (PAR) + 2

074CCC	SLT	SKIP IF LESS	CA	IF (A1)<(MEM(ADDR)) THEN PAR←(PAR)+2
075CCC	SNE	SKIP IF NOT EQUAL	CA	IF (A1)≠(MEM(ADDR)) THEN PAR←(PAR)+2.
076CCC	SLE	SKIP IF LESS OR EQUAL	CA	IF (A1)≤(MEM(ADDR)) THEN PAR←(PAR)+2
1020SS	LSA	LEFT SHIFT ARITHMETIC	AR	AQ[N]←AQ[N-1] (SIGN BIT NOT SHIFTED)
1040SS	LSL	LEFT SHIFT LOGICAL	AR	AQ[N]←(AQ[N-1]); Q1[1]←(A1[24])
1060SS	RSL	RIGHT SHIFT LOGICAL	AR	AQ[N]←(AQ[N+1]); A1[23]←(Q1[1])
110CCC	MPY	MULTIPLY	CA	AQ←(A1)*(MEM(ADDR))
112CCC	DIV	DIVIDE	CA	Q1←+(AQ/MEM(ADDR)); A1←[REMAINDER]; A1[24]←SIGN BIT OF QU
114000	NOP	NO-OP	OP	NO OPERATION
114001	ZAC	ZERO A1	OP	A1←0
114002	UAQ	UNITE A1 AND Q1	OP	Q1←(A1)∨(Q1)
114003	UOZ	UNITE A1 AND Q1, ZERO A1	OP	Q1←(A1)∨(Q1); A1←0
114004	ZQR	ZERO Q1	OP	Q1←0
114005	ZAQ	ZERO A1 AND Q1	OP	Q1←0; A1←0
114006	TAQ	TRANSFER A1 TO Q1	OP	Q1←(A1)
114007	MAZ	MOVE A1 TO Q1, ZERO A1	OP	Q1←(A1); A1←0
114011	TQA	TRANSFER Q1 TO A1	OP	A1←(Q1)
114040	OPC	OVERFLOW CHECK	OP	IF LAST INST. CAUSED OVERFLOW, SET ARITH. FLIP-FLOP
114100	ION	INTERRUPTS ON	OP	IE←1
114140	RSI	RESTORE INTERRUPTS	OP	IE←1; INT←0
114200	IOF	INTERRUPTS OFF	OP	IE←0
114400	HLT	HALT	OP	RUN←0
12CCCC	CXR	CONSTANT TO X REGISTER	DC	X[0:12]←(PCR[0:12]); X[13:23]←0
14CCCC	CXA	CONSTANT TO A1	DC	X[12:23]←(PCR[0:11]); A1←(X)
15CCCC	ACA	ADD CONSTANT TO A1	DC	X[12:23]←(PCR[0:11]); A1←(A1)+(X)
16DDDD	DWR	DRUM WRITE	DA	DRUM(ADDR)←(A1[0:16])
20DDDD	JMP	UNCONDITIONAL JUMP	DA	PAR←(PCR[0:12])
22DDDD	SNJ	SIGN JUMP	DA	IF (A1)<0 THEN PAR←(PCR[0:12])
24DDDD	NZJ	NON-ZERO JUMP	DA	IF (A1)≠0 THEN PAR←(PCR[0:12])
26DDDD	OFJ	OVERFLOW JUMP	DA	IF ARITH. FAULT THEN PAR←(PCR[0:12])
30DDDD	ZRJ	ZERO JUMP	DA	IF (A1)=0 THEN PAR←(PCR[0:12])
32DDDD	JMS	JUMP SUBROUTINE	DA	DRUM(ADDR)←(PAR)+BIT 16; PAR←(PCR[0:12])+1
34DDDD	WIO	WAIT FOR I/O	DA	RUN←0; WHEN PARTIAL SYNC SENT, PAR←(PCR[0:12]); RUN←1
360FFF	SKF	SKIP ON FLAGS	FL	IF FR[0:8]∧PCR[0:8]=PCR[0:8] THEN PAR←(PAR)+2
360000	SKP	SKIP (UNCONDITIONAL)	FL	PAR←(PAR)+2
362FFF	EXF	EXTRACT FLAGS	FL	FR[0:8]←(PCR[0:8])∧(FR[0:8])
364FFF	STF	SET FLAGS	FL	FR[0:8]←(PCR[0:8])∨(FR[0:8])
366FFF	LDF	LOAD FLAGS	FL	FR[0:8]←(PCR[0:8])
3700SS	SKS	SKIP ON SENSE SWITCHES	FL	IF SS[0:3]∧PCR[0:3]=PCR[0:3] THEN PAR←(PAR)+2
370040	SRC	SKIP ON READER CLEAR	FL	IF NO TAPE ERRORS ON LAST TAPE OPERATION THEN PAR←(PAR)+2
370100	SRS	SKIP IF READER SELECTED	FL	IF A READER IS SELECTED THEN PAR←(PAR)+2
370200	STR	SKIP ON TELETYPE READY	FL	IF TELETYPE READY, PAR←(PAR)+2
370400	SIF	SKIP ON INPUT FLAG	FL	IF IF=1 THEN PAR←(PAR)+2