MASTER COPY

L @GUIDE TO OP CODES RECOGNIZED BY YUL SYSTEM FOR BLK2 AND AGC USER'S OWN PAGE NO. 1 PAGE 1

0001 BY: HUGH BLAIR-SMITH IL7-238E MS73 XT 1215 OCTOBER 10, 1967

MIT/IC/KSC COPY

0002 THIS DOCUMENT LISTS ALPHABETICALLY ALL THE OPERATION CODES (CARD COLUMNS 18-23) RECOGNIZED BY THE YUL
0004 ASSEMBLERS FOR BLK2 AND AGC. EACH OP CODE IS ACCOMPANIED BY SOME VITAL STATISTICS: TYPE, ONE OR TWO REFERENCES
0006 (CITATIONS) TO FULLER DOCUMENTATION, EQUIVALENT NAME(S), SCOPE (DEGREE OF GENERALITY), MEANING OF APPENDED
0008 ASTERISK, AND THE SIGNIFICANCE OF THE ABBREVIATION. THERE IS ALSO A PAGE WITH SIMILAR INFORMATION ON ALL LEGAL
0010 CONTENTS OF COLUMNS 1 AND 8. DETAILED DESCRIPTIONS OF THE COLUMNS OF THE OP CODE GUIDE FOLLOW.

0012 OP CODE: THE ALPHABETIC ORDERING IS THE CLASSICAL PUNCHED-CARD COLLATING SEQUENCE: BLANK, SPECIAL CHARACTERS,
0014 ALPHABET, NUMERALS. EVERY OP CODE SHOWN WITH AN ASTERISK, E.G. BBCON*, IS ALSO LEGAL WITHOUT IT, E.G. BBCON.

0016 TYPE: OP CODES ARE CLASSIFIED INTO THE FOLLOWING TYPES:
0017 ADR CON (ADDRESS CONSTANT), ONE OR TWO WORDS MADE UP OF ONE OR MORE ADDRESSES BUT NO MACHINE OP CODES;
0019 CLERICAL, AN INSTRUCTION TO THE ASSEMBLER RATHER THAN TO THE AGC;
0020 INSTR (INSTRUCTION), A WORD DIRECTLY EXECUTABLE BY THE AGC;
0021 MERGE (MERGE CONTROL CARD), AN INSTRUCTION TO THE FILE-UPDATING PART OF THE YUL SYSTEM, GOVERNING MERGING
0023 OF INPUT CARDS WITH OLD PROGRAM;
0024 NUM CON (NUMERIC CONSTANT), ONE OR TWO WORDS REPRESENTING A NUMBER;
0025 PRET OP (INTERPRETIVE OP CODE), PACKED 2 TO A WORD IN INTERPRETIVE PROGRAMS; AND
0027 OBSOLETE, ONE CODE (MODIFY) WHICH IS RECOGNIZED BY YUL-1800, BUT DOESN'T DO ANYTHING USEFUL.

0029 REF/PAGE: THE FOLLOWING REFERENCES ARE CITED:
0030 D (DIGEST), HUGH BLAIR-SMITH, "A YUL PROGRAMMER'S DIGEST," REVISION 7, AUGUST 31, 1967 (COMPUTER PRINTOUT);
0032 G (GUIDE), THIS DOCUMENT;
0033 I (INTERPRETER BOOK), CHARLES A. MUNTZ, "USERS GUIDE TO THE BLOCK II AGC/LGC INTERPRETER," APRIL 1965 (R-489);
0035 M (MACHINE BOOK), HUGH BLAIR-SMITH, "AGC4 MEMO #9 - BLOCK II INSTRUCTIONS," REVISED JUNE 1, 1967;
0037 T (TIPS), HUGH BLAIR-SMITH, "YUL TIPS," PUBLISHED FROM TIME TO TIME IN "THE SUNLITER" (BARBARA JO JONTZ, ED.);
0039 Y (YUL BOOK), RAYTHEON AGC INFORMATION SERIES ISSUE 13, "YUL PROGRAMMING SYSTEM," 5 DECEMBER 1963.
0041 EACH CITATION IS GIVEN IN THE FORM R/PP; I.E. Y/80 MEANS PAGE 80 OF REFERENCE Y. IN MOST OF THE CITATIONS OF
0043 THE DIGEST, A SPECIAL SERIES OF PAGE NUMBERS IS USED: C_1, C_2, C_3, C_4. EACH OF THESE REFERS TO TWO PLACES AT
0045 ONCE, ONE IN THE SECTION "AGC BLOCK II INTERPRETIVE CODES: ASSEMBLY RULES FOR BLK2" AND ONE IN THE SECTION
0047 "AGC BLOCK II INTERPRETIVE CODES: ASSEMBLY RULES FOR AGC." FOR EXAMPLE, C2 REFERS TO "USER'S OWN PAGE NO. 2"
0049 IN BOTH SECTIONS. THIS DEVICE HELPS IN FINDING THE DIFFERENCES BETWEEN THE TWO ASSEMBLERS. THE "PAGE NUMBER"
0051 IN A YUL TIP CITATION IS REALLY THE SERIAL NUMBER OF THE TIP. WHERE TWO CITATIONS ARE GIVEN, THE FIRST IS THE
0053 BETTER OR MORE BASIC ONE.

0054 SAME AS: AN OP CODE IN THIS COLUMN HAS EXACTLY THE SAME EFFECT ON THE ASSEMBLER AS THE ONE ON THE LEFT, THOUGH
0056 PROGRAMMING CONVENTIONS SOMETIMES MAKE AN ARTIFICIAL DISTINCTION, AS BETWEEN "EQUALS" AND "=" IN DISKERAS ETC.
0058 WHEN AN OP CODE IN THIS COLUMN IS DEFINITELY A LESS PREFERRED ONE, IT IS ENCLOSED IN ANGLE BRACKETS <THUS>.

0060 SCOPE: AN OP CODE MARKED "GEN" IS GENERAL, THAT IS, THE SAME FOR ALL COMPUTERS SERVED BY YUL-1800, EXCEPT
0062 INSOFAR AS PARAMETERS SUCH AS WORD LENGTH MAKE A DIFFERENCE. IF IT IS MARKED "SAME," IT IS UNIQUE TO THE BLOCK
0064 II AGC, BUT THE SAME IN THE BLK2 AND AGC ASSEMBLERS. "DIFF" MEANS THAT THE OP CODE IS TREATED DIFFERENTLY BY
0066 THE BLK2 AND AGC ASSEMBLERS, IN WAYS THAT CAN BE FOUND THROUGH THE CITATIONS.

0068 EFFECT OF *: THE OPTIONAL ASTERISK EITHER MEANS INTERPRETIVE INDEXING OR IT HAS SOME SPECIAL MEANING THAT CAN
0070 BE FOUND THROUGH THE CITATION IN THIS COLUMN.

0071 SIGNIFICANCE OF ABBREVIATION: USUALLY, THE PURPOSE OF THIS COLUMN IS SIMPLY TO SHOW WHICH WORDS WERE COM-
0073 PRESSED, BENT, FOLDED, AND MUTILATED TO FORM THE OP CODE. WHENEVER THIS YIELDS LITTLE MEANING, ESPECIALLY IN
0075 THE RARELY USED CODES, THERE IS ADDITIONAL INFORMATION. HOWEVER, THIS COLUMN IS NO SUBSTITUTE FOR THE CITED
0077 REFERENCES FOR FINDING OUT WHAT A GIVEN OP CODE REALLY DOES.

0078	OP CODE	TYPE	REF/PAGE	SAME AS	SCOPE	EFFECT OF *	SIGNIFICANCE OF ABBREVIATION
0080	-------	----	--------	-------	-----	-----------	----------------------------
0082	(BLANK)	ADR CON	G/10 I/6	<P>	DIFF		(MOST STRAIGHTFORWARD ADDRESS TYPE)
0084	=	CLERICAL	Y/80 G/10	EQUALS,<IS>	GEN		EQUALS
0085	=MINUS	CLERICAL	T/10 G/11		"		EQUALS MINUS LOCATION COUNTER
0087	=PLUS	CLERICAL	T/10 G/11		"		EQUALS PLUS LOCATION COUNTER
0089	ABS	PRET OP	I/66 D/C3		SAME		ABSOLUTE VALUE (SCALAR)
0091	ABVAL	PRET OP	I/66 D/C3		"		ABSOLUTE VALUE (VECTOR)
0093	ACOS	PRET OP	I/66 D/C3	ARCCOS	"		ARC COSINE
0094	AD	INSTR	M/10		"		ADD
0095	ADRES	ADR CON	M/25		"		ADDRESS
0096	ADS	INSTR	M/9		"		ADD TO STORAGE
0098	ARCCOS	PRET OP	I/10 D/C3	ACOS	"		ARC COSINE
0099	ARCSIN	PRET OP	I/10 D/C3	ASIN	"		ARC SINE
0100	ASIN	PRET OP	I/66 D/C3	ARCSIN	"		ARC SINE
0101	AUG	INSTR	M/13		"		AUGMENT
0102	AXC,1	PRET OP	I/66 D/C2		"		ADDRESS TO INDEX 1 COMPLEMENTED
0104	AXC,2	PRET OP	I/66 D/C2		"		ADDRESS TO INDEX 2 COMPLEMENTED
0106	AXT,1	PRET OP	I/66 D/C2		"		ADDRESS TO INDEX 1 TRUE
0108	AXT,2	PRET OP	I/66 D/C2		"		ADDRESS TO INDEX 2 TRUE
0110	BANK	CLERICAL	G/11		SAME		SEEK FREE LOCATION IN THIS BANK
0112	BBCON*	ADR CON	M/27 G/11		DIFF	SEE G/11	BOTH-BANK CONSTANT
0114	BDDV*	PRET OP	I/66 D/C1		SAME	INDEXING	BACKWARDS DOUBLE DIVIDE
0116	BDSU*	PRET OP	I/66 D/C1		"	INDEXING	BACKWARDS DOUBLE SUBTRACT
0118	BHIZ	PRET OP	I/66 D/C2		DIFF		BRANCH ON HIGH-ORDER ZERO
0120	BLOCK	CLERICAL	G/12 Y/77		GEN		SEEK FREE LOCATION IN THIS BLOCK
0122	BMN	PRET OP	I/66 D/C2		SAME		BRANCH ON MINUS
0124	BNKSUM	ADR CON	G/12		DIFF		GENERATE BANK SUM PATTERN FOR SELFCHECK
0126	BOF	PRET OP	D/C3	BOFF	SAME		BRANCH IF BIT OFF
0128	BOFCLR	PRET OP	I/66 D/C3		"		BRANCH IF BIT OFF, CLEAR BIT
0130	BOFF	PRET OP	I/66 D/C3	BOF	"		BRANCH IF BIT OFF
0132	BOFINV	PRET OP	I/66 D/C3		"		BRANCH IF BIT OFF, INVERT BIT
0134	BOFSET	PRET OP	I/66 D/C3		"		BRANCH IF BIT OFF, SET BIT
0136	BON	PRET OP	I/66 D/C3		"		BRANCH IF BIT ON
0138	BONCLR	PRET OP	I/66 D/C3		"		BRANCH IF BIT ON, CLEAR BIT
0140	BONINV	PRET OP	I/66 D/C3		"		BRANCH IF BIT ON, INVERT BIT
0142	BONSET	PRET OP	I/66 D/C3		"		BRANCH IF BIT ON, SET BIT
0144	BOV	PRET OP	I/66 D/C2		"		BRANCH ON OVERFLOW
0146	BOVB	PRET OP	I/66 D/C2		"		BRANCH ON OVERFLOW TO BASIC
0148	BPL	PRET OP	I/66 D/C2		"		BRANCH ON PLUS
0150	BVSU*	PRET OP	I/66 D/C1		"	INDEXING	BACKWARDS VECTOR SUBTRACT
0152	BZE	PRET OP	I/66 D/C2		"		BRANCH ON ZERO
0154	BZF	INSTR	M/13		"		BRANCH ON ZERO TO FIXED
0156	BZMF	INSTR	M/14		"		BRANCH ON ZERO OR MINUS TO FIXED

0158	OP CODE	TYPE	REF/PAGE	SAME AS	SCOPE	EFFECT OF *	SIGNIFICANCE OF ABBREVIATION
0160	-------	----	--------	-------	-----	----------	----------------------------
0162	CA	INSTR	M/9		SAME		CLEAR AND ADD
0164	CADR	ADR CON	M/26	FCADR	"		COMPLETE ADDRESS
0166	CAE	INSTR	M/9	CA	"		CLEAR AND ADD FROM ERASABLE
0168	CAF	INSTR	M/9	CA	"		CLEAR AND ADD FROM FIXED
0170	CALL	PRET OP	I/67 D/C_2		DIFF		CALL
0171	CALRB	PRET OP	D/C2		"		CALL AND RETURN IN BASIC
0173	CARDNO	MERGE	Y/86 D/8		GEN		CARD NUMBER
0174	CARDNS	MERGE	D/8		"		CARD NUMBERS
0175	CCALL*	PRET OP	I/67 D/C_1		DIFF	INDEXING	COMPUTED CALL
0177	CCLRB*	PRET OP	D/C1		SAME	INDEXING	COMPUTED CALL AND RETURN IN BASIC
0179	CCS	INSTR	M/8		"		COUNT, COMPARE AND SKIP
0181	CGOTO*	PRET OP	I/67 D/C_1		"	INDEXING	COMPUTED GO TO
0183	CLEAR	PRET OP	I/67 D/C_3	CLR	"		CLEAR A BIT
0184	CLR	PRET OP	D/C3	CLEAR	"		CLEAR A BIT
0185	CLRGO	PRET OP	I/67 D/C_3		"		CLEAR A BIT AND GO TO
0187	COM	INSTR	M/19		"		COMPLEMENT
0188	COS	PRET OP	I/67 D/C_3	COSINE	"		COSINE
0189	COSINE	PRET OP	I/10 D/C_3	COS	"		COSINE
0190	COUNT*	CLERICAL	T/8 T/10		GEN	SEE T/10	COUNT WORDS
0191	CS	INSTR	M/9		SAME		CLEAR AND SUBTRACT
0193	DAD*	PRET OP	I/67 D/C_1		SAME	INDEXING	DOUBLE ADD
0194	DAS	INSTR	M/8		"		DOUBLE ADD TO STORAGE
0196	DCA	INSTR	M/13		"		DOUBLE CLEAR AND ADD
0198	DCOM	INSTR	M/21		"		DOUBLE COMPLEMENT
0200	DCOMP	PRET OP	I/67 D/C_3		"		DOUBLE COMPLEMENT
0202	DCS	INSTR	M/14		"		DOUBLE CLEAR AND SUBTRACT
0204	DDOUBL	INSTR	M/16		"		DOUBLE DOUBLE
0206	DDV*	PRET OP	I/67 D/C_1		"	INDEXING	DOUBLE DIVIDE
0208	DEC*	NUM CON	Y/67		GEN	SEE Y/69	DECIMAL CONSTANT
0210	DELETE	MERGE	Y/85		"		DELETE CARDS
0211	DIM	INSTR	M/13		SAME		DIMINISH
0212	DLOAD*	PRET OP	I/67 D/C_1		"	INDEXING	DOUBLE LOAD
0213	DMP*	PRET OP	I/67 D/C_1		"	INDEXING	DOUBLE MULTIPLY
0215	DMPR*	PRET OP	I/67 D/C_1		"	INDEXING	DOUBLE MULTIPLY AND ROUND
0217	DNCHAN	ADR CON	M/26.1		"		DOWNLINK CHANNEL ADDRESS
0219	DNPTR	ADR CON	M/26.1		"		DOWNLINK LIST POINTER
0221	DOT*	PRET OP	I/67 D/C_1		"	INDEXING	DOT PRODUCT
0222	DOUBLE	INSTR	M/20		"		DOUBLE
0223	DSQ	PRET OP	I/67 D/C_3		"		DOUBLE SQUARE
0225	DSU*	PRET OP	I/67 D/C_1		"	INDEXING	DOUBLE SUBTRACT
0227	DTCB	INSTR	M/20		"		DOUBLE TRANSFER CONTROL, SETTING BOTH BANKS
0229	DTCF	INSTR	M/19		"		DOUBLE TRANSFER CONTROL, SETTING FBANK
0231	DV	INSTR	M/12		"		DIVIDE
0232	DXCH	INSTR	M/10		"		DOUBLE EXCHANGE

0234	OP CODE	TYPE	REF/PAGE	SAME AS	SCOPE	EFFECT OF *	SIGNIFICANCE OF ABBREVIATION
0236							
0238	EBANK=	CLERICAL	M/26 G/13		DIFF		E BANK SETTING EQUALS
0240	ECADR	ADR CON	M/26	1DNADR	SAME		ERASABLE-MEMORY COMPLETE ADDRESS
0242	EDRUPT	INSTR	M/12 G/13		DIFF		E SMALLY'S INTERRUPT INSTRUCTION
0244	EQUALS	CLERICAL	Y/80 G/10	=,<IS>	GEN		EQUALS
0245	ERASE	CLERICAL	Y/78		"		RESERVE ERASABLE MEMORY
0247	EVEN	CLERICAL	G/14		"		MAKE LOCATION COUNTER EVEN
0249	EXIT	PRET OP	I/67 D/C3		SAME		EXIT
0250	EXTEND	INSTR	M/16		"		EXTEND
0251	FCADR	ADR CON	M/26	CADR	SAME		FIXED-MEMORY COMPLETE ADDRESS
0253	GENADR	ADR CON	M/25		SAME		GENERALIZED ADDRESS
0255	GO TO	PRET OP	D/C2	GOTO	"		GO TO
0256	GOTO	PRET OP	I/67 D/C2	GO TO	"		GO TO
0257	HEAD	CLERICAL	Y/82	<TAIL>	GEN		ESTABLISH HEAD CHARACTER
0259	INCR	INSTR	M/9		SAME		INCREMENT
0260	INCR,1	PRET OP	I/67 D/C2		"		INCREMENT INDEX 1
0262	INCR,2	PRET OP	I/67 D/C2		"		INCREMENT INDEX 2
0264	INDEX	INSTR	M/10 M/14	<NDX>	"		INDEX
0265	INHINT	INSTR	M/16		"		INHIBIT INTERRUPT
0267	INSERT	MERGE	T/10		GEN		INSERT CARDS
0268	INV	PRET OP	D/C3	INVERT	SAME		INVERT A BIT
0269	INVERT	PRET OP	I/67 D/C3	INV	"		INVERT A BIT
0270	INVGO	PRET OP	I/67 D/C3		"		INVERT A BIT AND GO TO
0272	IS	CLERICAL	Y/80 G/10	EQUALS,=	GEN		IS EQUIVALENT TO
0274	ITA	PRET OP	I/15 D/C2	STQ	DIFF		INTERPRETIVE TRANSFER ADDRESS TO STORAGE
0276	ITCQ	PRET OP	I/15 D/C3	RVQ	SAME		INTERPRETIVE TRANSFER CONTROL TO Q
0278	LOC	CLERICAL	Y/76	SETLOC	GEN		SET LOCATION COUNTER
0280	LXA,1	PRET OP	I/67 D/C2		SAME		LOAD INDEX 1 FROM ADDRESS
0282	LXA,2	PRET OP	I/67 D/C2		"		LOAD INDEX 2 FROM ADDRESS
0284	LXC,1	PRET OP	I/67 D/C2		"		LOAD INDEX 1 FROM ADDRESS COMPLEMENTED
0286	LXC,2	PRET OP	I/67 D/C2		"		LOAD INDEX 2 FROM ADDRESS COMPLEMENTED
0288	LXCH	INSTR	M/9		"		L EXCHANGE

0289	OP CODE	TYPE	REF/PAGE	SAME AS	SCOPE	EFFECT OF *	SIGNIFICANCE OF ABBREVIATION
0291							
0293	MASK	INSTR	M/11	<MSK>	SAME		MASK
0294	MEMORY	CLERICAL	Y/82		GEN		MEMORY TYPE
0295	MM	NUM CON	G/14		SAME		MAJOR MODE CONSTANT
0297	MODIFY	OBSOLETE	G/14		GEN		MODIFY SUBROUTINE
0299	MP	INSTR	M/14		SAME		MULTIPLY
0300	MSK	INSTR	M/11	MASK	"		MASK
0301	MSU	INSTR	M/13		"		MODULAR SUBTRACT
0303	MXV*	PRET OP	I/67 D/C1		"	INDEXING	MATRIX TIMES VECTOR
0305	NDX	INSTR	M/7	INDEX	SAME		INDEX
0306	NOOP	INSTR	M/16 M/19		"		NO OPERATION
0307	NORM*	PRET OP	I/67 D/C1	<SLC*>	"	INDEXING	NORMALIZE
0308	NV	NUM CON	G/14	VN	"		NOUN-VERB CONSTANT
0310	OCT*	NUM CON	Y/67	<OCTAL*>	GEN	SEE Y/69	OCTAL CONSTANT
0312	OCTAL*	NUM CON	Y/67	OCT*	"	SEE Y/69	OCTAL CONSTANT
0314	OVSK	INSTR	M/20		SAME		OVERFLOW SKIP
0316	P	ADR CON	G/10	(BLANK)	SAME		POLISH ADDRESS
0318	PDDL*	PRET OP	I/68 D/C1		"	INDEXING	PUSH DOWN AND DOUBLE LOAD
0320	PDVL*	PRET OP	I/68 D/C1		"	INDEXING	PUSH DOWN AND VECTOR LOAD
0322	PUSH	PRET OP	I/68 D/C3		"		PUSH DOWN
0323	QXCH	INSTR	M/13		SAME		Q EXCHANGE
0324	RAND	INSTR	M/11		SAME		READ, AND
0325	READ	INSTR	M/11		"		READ
0326	RELINT	INSTR	M/16		"		RELEASE INTERRUPT
0328	REMADR	ADR CON	M/25		"		REMOTE ADDRESS
0330	RESUME	INSTR	M/19		"		RESUME
0331	RETURN	INSTR	M/15		"		RETURN
0332	ROR	INSTR	M/11		"		READ, OR
0333	ROUND	PRET OP	I/68 D/C3		"		ROUND
0334	RTB	PRET OP	I/68 D/C2		DIFF		RETURN TO BASIC
0336	RVQ	PRET OP	I/68 D/C3	<ITCQ>	SAME		RETURN VIA QPRET
0338	RXOR	INSTR	M/12		"		READ, EXCLUSIVE OR

0340	OP CODE	TYPE	REF/PAGE	SAME AS	SCOPE	EFFECT OF *	SIGNIFICANCE OF ABBREVIATION
0342	---	---	---	---	---	---	---
0344	SBANK*	CLERICAL	M/26.1		SAME		SUPER-BANK SETTING EQUALS
0346	SET	PRET OP	I/68 D/C3		"		SET A BIT
0347	SETGO	PRET OP	I/68 D/C3		"		SET A BIT AND GO TO
0349	SETLOC	CLERICAL	Y/76	<LOC>	GEN		SET LOCATION COUNTER
0351	SETPD	PRET OP	I/68 D/C1		SAME		SET PUSH-DOWN POINTER
0353	SIGN*	PRET OP	I/68 D/C1		"	INDEXING	SIGN
0354	SIN	PRET OP	I/68 D/C3	SINE	"		SINE
0355	SINE	PRET OP	I/10 D/C3	SIN	"		SINE
0356	SL*	PRET OP	I/68 D/C1		"	INDEXING	SHIFT LEFT
0357	SLC*	PRET OP	I/13 D/C1	NORM*	"	INDEXING	SHIFT LEFT AND COUNT
0359	SLOAD*	PRET OP	I/68 D/C1		"	INDEXING	SINGLE LOAD
0360	SLR*	PRET OP	I/68 D/C1		"	INDEXING	SHIFT LEFT AND ROUND
0362	SL1	PRET OP	I/68 D/C4		"		SHIFT LEFT 1
0363	SL1R	PRET OP	I/68 D/C4		"		SHIFT LEFT 1 AND ROUND
0365	SL2	PRET OP	I/68 D/C4		"		SHIFT LEFT 2
0366	SL2R	PRET OP	I/68 D/C4		"		SHIFT LEFT 2 AND ROUND
0368	SL3	PRET OP	I/68 D/C4		"		SHIFT LEFT 3
0369	SL3R	PRET OP	I/68 D/C4		"		SHIFT LEFT 3 AND ROUND
0371	SL4	PRET OP	I/68 D/C4		"		SHIFT LEFT 4
0372	SL4R	PRET OP	I/68 D/C4		"		SHIFT LEFT 4 AND ROUND
0374	SQUARE	INSTR	M/21		"		SQUARE
0375	SQRT	PRET OP	I/68 D/C3		"		SQUARE ROOT
0376	SR*	PRET OP	I/68 D/C1		"	INDEXING	SHIFT RIGHT
0377	SRR*	PRET OP	I/68 D/C1		"	INDEXING	SHIFT RIGHT AND ROUND
0379	SR1	PRET OP	I/68 D/C4		"		SHIFT RIGHT 1
0381	SR1R	PRET OP	I/68 D/C4		"		SHIFT RIGHT 1 AND ROUND
0383	SR2	PRET OP	I/68 D/C4		"		SHIFT RIGHT 2
0385	SR2R	PRET OP	I/68 D/C4		"		SHIFT RIGHT 2 AND ROUND
0387	SR3	PRET OP	I/68 D/C4		"		SHIFT RIGHT 3
0389	SR3R	PRET OP	I/68 D/C4		"		SHIFT RIGHT 3 AND ROUND
0391	SR4	PRET OP	I/68 D/C4		"		SHIFT RIGHT 4
0393	SR4R	PRET OP	I/68 D/C4		"		SHIFT RIGHT 4 AND ROUND
0395	SSP*	PRET OP	I/69 D/C1		"	INDEXING	SET SINGLE PRECISION
0397	STADR	PRET OP	I/69 D/C3		"		STORE ADDRESS COMING
0399	STCALL	ADR CON	I/69 D/C4		DIFF		STORE AND CALL
0401	STODL*	ADR CON	I/69 D/C4		"	INDEXING	STORE AND DOUBLE LOAD
0403	STORE	ADR CON	I/69 D/C4		"		STORE
0404	STOVL*	ADR CON	I/69 D/C4		"	INDEXING	STORE AND VECTOR LOAD
0406	STQ	PRET OP	I/69 D/C2	<ITA>	"		STORE QPRET
0407	SU	INSTR	M/14		SAME		SUBTRACT
0408	SUBRO*	CLERICAL	Y/83 G/14		GEN	SEE G/14	SUBROUTINE
0409	SXA,1	PRET OP	I/69 D/C2		SAME		STORE INDEX 1 IN ADDRESS
0411	SXA,2	PRET OP	I/69 D/C2		"		STORE INDEX 2 IN ADDRESS

0413 / 0415	OP CODE	TYPE	REF/PAGE	SAME AS	SCOPE	EFFECT OF *	SIGNIFICANCE OF ABBREVIATION
0417	TAD*	PRET OP	I/69 D/C1		SAME	INDEXING	TRIPLE ADD
0418	TAIL	CLERICAL	Y/82	HEAD	GEN		ESTABLISH TAIL (HEAD) CHARACTER
0420	TC	INSTR	M/3	TCR	SAME		TRANSFER CONTROL
0422	TCAA	INSTR	M/20		"		TRANSFER CONTROL TO ADDRESS IN A
0424	TCF	INSTR	M/8		"		TRANSFER CONTROL TO FIXED
0426	TCR	INSTR	M/3	TC	"		TRANSFER CONTROL, SETTING UP RETURN
0428	TIX,1	PRET OP	I/69 D/C2		"		TRANSFER CONTROL IF INDEX 1 IS DECREMENTABLE
0430	TIX,2	PRET OP	I/69 D/C2		"		TRANSFER CONTROL IF INDEX 2 IS DECREMENTABLE
0432	TLOAD*	PRET OP	I/69 D/C1		"	INDEXING	TRIPLE LOAD
0433	TS	INSTR	M/10		"		TRANSFER TO STORAGE
0435	UNIT	PRET OP	I/69 D/C3		SAME		UNIT VECTOR
0436	V/SC*	PRET OP	I/70 D/C1		SAME	INDEXING	VECTOR OVER SCALAR
0438	VAD*	PRET OP	I/69 D/C1		"	INDEXING	VECTOR ADD
0439	VCOMP	PRET OP	I/69 D/C3		"		VECTOR COMPLEMENT
0441	VDEF	PRET OP	I/69 D/C3		"		VECTOR DEFINE
0443	VLOAD*	PRET OP	I/69 D/C1		"	INDEXING	VECTOR LOAD
0444	VN	NUM CON	G/14	<NV>	"		VERB-NOUN CONSTANT
0446	VPROJ*	PRET OP	I/69 D/C1		"	INDEXING	VECTOR PROJECT
0448	VSL*	PRET OP	I/69 D/C1		"	INDEXING	VECTOR SHIFT LEFT
0450	VSL1	PRET OP	I/69 D/C4		"		VECTOR SHIFT LEFT 1
0452	VSL2	PRET OP	I/69 D/C4		"		VECTOR SHIFT LEFT 2
0454	VSL3	PRET OP	I/69 D/C4		"		VECTOR SHIFT LEFT 3
0456	VSL4	PRET OP	I/69 D/C4		"		VECTOR SHIFT LEFT 4
0458	VSL5	PRET OP	I/69 D/C4		"		VECTOR SHIFT LEFT 5
0460	VSL6	PRET OP	I/69 D/C4		"		VECTOR SHIFT LEFT 6
0462	VSL7	PRET OP	I/69 D/C4		"		VECTOR SHIFT LEFT 7
0464	VSL8	PRET OP	I/69 D/C4		"		VECTOR SHIFT LEFT 8
0466	VSQ	PRET OP	I/70 D/C3		"		VECTOR SQUARE
0468	VSR*	PRET OP	I/69 D/C1		"	INDEXING	VECTOR SHIFT RIGHT
0470	VSR1	PRET OP	I/69 D/C4		"		VECTOR SHIFT RIGHT 1
0472	VSR2	PRET OP	I/69 D/C4		"		VECTOR SHIFT RIGHT 2
0474	VSR3	PRET OP	I/69 D/C4		"		VECTOR SHIFT RIGHT 3
0476	VSR4	PRET OP	I/69 D/C4		"		VECTOR SHIFT RIGHT 4
0478	VSR5	PRET OP	I/69 D/C4		"		VECTOR SHIFT RIGHT 5
0480	VSR6	PRET OP	I/70 D/C4		"		VECTOR SHIFT RIGHT 6
0482	VSR7	PRET OP	I/70 D/C4		"		VECTOR SHIFT RIGHT 7
0484	VSR8	PRET OP	I/70 D/C4		"		VECTOR SHIFT RIGHT 8
0486	VSU*	PRET OP	I/70 D/C1		"	INDEXING	VECTOR SUBTRACT
0488	VXM*	PRET OP	I/70 D/C1		"	INDEXING	VECTOR TIMES MATRIX
0490	VXSC*	PRET OP	I/70 D/C1		"	INDEXING	VECTOR TIMES SCALAR
0492	VXV*	PRET OP	I/70 D/C1		"	INDEXING	VECTOR CROSS VECTOR

	OP CODE	TYPE	REF/PAGE	SAME AS	SCOPE	EFFECT OF *	SIGNIFICANCE OF ABBREVIATION
0498	WAND	INSTR	M/11		SAME		WRITE, AND
0499	WOR	INSTR	M/12		"		WRITE, OR
0500	WRITE	INSTR	M/11		"		WRITE
0501	XAD,1	PRET OP	I/70 D/C2		SAME		INDEX 1 ADD
0502	XAD,2	PRET OP	I/70 D/C2		"		INDEX 2 ADD
0503	XCH	INSTR	M/10		"		EXCHANGE
0504	XCHX,1	PRET OP	I/70 D/C2		"		EXCHANGE INDEX 1
0506	XCHX,2	PRET OP	I/70 D/C2		"		EXCHANGE INDEX 2
0508	XLQ	INSTR	M/15		"		EXECUTE USING L AND Q
0510	XSU,1	PRET OP	I/70 D/C2		"		INDEX 1 SUBTRACT
0512	XSU,2	PRET OP	I/70 D/C2		"		INDEX 2 SUBTRACT
0514	XXALQ	INSTR	M/15		"		EXECUTE EXTRACODE USING A, L, AND Q
0516	ZL	INSTR	M/19		SAME		ZERO L
0517	ZQ	INSTR	M/20		"		ZERO Q
0518	0	INSTR	G/15	SEE M/34	SAME		000 BINARY
0519	1	INSTR	G/15	SEE M/34	SAME		001 BINARY
0520	1DNADR	ADR CON	M/26,1	ECADR	"		1-DP-WORD-FOR-DOWNLINK ADDRESS
0522	2	INSTR	G/15	SEE M/34	SAME		010 BINARY
0523	2BCADR	ADR CON	M/27 G/15	2CADR	DIFF		2-WORD BBANK-TYPE COMPLETE ADDRESS
0525	2CADR	ADR CON	M/27 G/15	2BCADR	"		2-WORD COMPLETE ADDRESS
0527	2DEC*	NUM CON	Y/67		GEN	SEE Y/69	2-WORD DECIMAL CONSTANT
0529	2DNADR	ADR CON	M/26,1		SAME		2-DP-WORD-FOR-DOWNLINK ADDRESS
0531	2FCADR	ADR CON	M/27		"		2-WORD FBANK-TYPE COMPLETE ADDRESS
0533	2OCT*	NUM CON	Y/67	<2OCTAL>	GEN	SEE Y/69	2-WORD OCTAL CONSTANT
0535	2OCTAL	NUM CON	Y/67	2OCT	"		2-WORD OCTAL CONSTANT
0537	3	INSTR	G/15	SEE M/34	SAME		011 BINARY
0538	3DNADR	ADR CON	M/26,1		"		3-DP-WORD-FOR-DOWNLINK ADDRESS
0540	4	INSTR	G/15	SEE M/34	SAME		100 BINARY
0541	4DNADR	ADR CON	M/26,1		"		4-DP-WORD-FOR-DOWNLINK ADDRESS
0543	5	INSTR	G/15	SEE M/34	SAME		101 BINARY
0544	5DNADR	ADR CON	M/26,1		"		5-DP-WORD-FOR-DOWNLINK ADDRESS
0546	6	INSTR	G/15	SEE M/34	SAME		110 BINARY
0547	6DNADR	ADR CON	M/26,1		"		6-DP-WORD-FOR-DOWNLINK ADDRESS
0549	7	INSTR	G/15	SEE M/34	SAME		111 BINARY

GUIDE TO COLUMN 1 AND 8 CONTENTS MEANINGFUL TO YUL

COLUMN 1	TYPE	REF/PAGE	SIGNIFICANCE OF ABBREVIATION
(BLANK)	(SEE OP)	G/9	COMMONEST TYPE
=	MERGE	D/7 Y/87	SEEK AND ACCEPT A CARD EQUAL TO THIS ONE
*	MONITOR	Y/51 G/9	IBM-7090 FAP MONITOR HAD * ON MONITOR CARDS
A	REMARK	Y/51 Y/90	ALIGNED REMARKS (ALIGNED WITH INSTRUCTIONS)
J	LEFTOVER	Y/72 G/9	JOKER (WILD CARD)--VALUE ASSIGNED BY YUL. NOTE: OP CODE TYPE MUST BE INSTR OR CONST
L	LOG	Y/88 D/5	LOG CARD. INVENTED IN IBM-650 DAYS TO PRINT LOG NUMBER AND PAGE HEADING
P	REMARK	Y/51 Y/90	PAGE-HEAD REMARKS
R	REMARK	Y/51 Y/89	REMARKS
S	SUBDIR	Y/49 D/4	SUBDIRECTOR--MEANINGLESS UNLESS FOLLOWING A YUL DIRECTOR CARD (Q.V.)
T	MERGE	Y/81 D/8	TUCK IN A NEW SECTION (AHEAD OF NEXT LOG)
Y	DIRECTOR	Y/36 D/2	YUL DIRECTOR--FIRST CARD OF EVERY YUL TASK

COLUMN 8	MEANS	REF/PAGE
(BLANK)	SPACE 1	Y/55
0	SPACE 0	Y/55
1	SPACE 1	Y/55
2	SPACE 2	Y/55
3	SPACE 3	Y/55
4	SPACE 4	Y/55
5	SPACE 5	Y/55
6	SPACE 6	Y/55
7	SPACE 7	Y/55
8	SKIP	Y/55
9	RT PRINT	Y/55 Y/90

NOTES ON THIS PAGE

BLANK COLUMN 1: THE DECODING OF THIS TYPE IS GOVERNED BY THE OP CODE IN COLUMNS 18-23 UNLESS COLUMN 8 CONTAINS A 9, IN WHICH CASE THE CARD IS A RIGHT-PRINT REMARKS CARD, IRRESPECTIVE OF COLUMNS 18-23. IN FACT, A 9 IN COLUMN 8 WILL OVERPOWER ANY COLUMN 1 CHARACTER EXCEPT *, S, AND Y.

* IN COLUMN 1: EITHER A MONITOR CARD OR A Y CARD MARKS THE END OF INPUT TO ANY YUL TASK. SYSTEM/360 MONITOR CARDS BEGIN WITH EITHER // OR /*. IN ANY CASE, MONITOR CARDS ARE SUPPLIED BY THE OPERATOR.

J IN COLUMN 1: THE LEFTOVER RULES GIVEN AT Y/72 HAVE BEEN REFINED TO MAKE THE LEFTOVER GO INTO THE RIGHT BANK, AS FAR AS THIS CAN BE DETERMINED. THE DETERMINATION IS BASED ON THE STATE OF THE LOCATION COUNTER AT THE TIME THE LEFTOVER IS ENCOUNTERED, AS FOLLOWS:

LOCATION COUNTER:	UNDETERMINED	UNSWITCHED ERAS.	ANY E-BANK E3-E7	FIXED-FIXED	ANY F-BANK 00-43
LEFTOVER WORD(S):	ANYWHERE IN FIXED	ANYWHERE IN FIXED	ANYWHERE IN FIXED	IN FIXED-FIXED	IN SAME F-BANK
LEFTOVER ERASE:	ANYWHERE IN ERAS.	IN UNSWITCHED E	IN SAME E-BANK	ANYWHERE IN ERAS.	ANYWHERE IN ERAS.

0602 NOTES ON SOME OP CODES
0603 ----------

0604 (BLANK), <P>

0605 THIS IS THE CODE FOR INTERPRETIVE ADDRESS CONSTANTS. REFER TO D/C1-D/C4 FOR THE NUMBER AND KINDS OF
0607 ADDRESSES REQUIRED BY EACH INTERPRETIVE OP CODE. THIS OP CODE IS REQUIRED FOR ALL KINDS OF ADDRESSES EXCEPT C;
0609 C ADDRESSES ARE REALLY OPERANDS AND MAY HAVE THIS OR ANY OTHER OP CODE THAT GENERATES A ONE-WORD CONSTANT OR
0611 INSTRUCTION.

0612 =, EQUALS, <IS>

0613 THE RULES GOVERNING ADDRESS SYMBOLS NOT PREDEFINED IN PASS 1, GIVEN AT Y/81, HAVE BEEN IMPROVED. THE
0615 CURRENT RULE IS THIS: IF THE ADDRESS SYMBOL HAS NOT BEEN PREDEFINED WHEN THE EQUALS CARD REFERRING TO IT IS
0617 ENCOUNTERED IN PASS 1, THEN THE LOCATION SYMBOL IS STORED IN THE SYMBOL TABLE WITH A POINTER TO THE ADDRESS
0619 SYMBOL'S PLACE IN THE TABLE (DEFINER THREAD) AND A 16-BIT FIELD CONTAINING THE SIGNED MODIFIER FROM THE ADDRESS
0621 FIELD, AND THE ADDRESS SYMBOL IS STORED IN THE SYMBOL TABLE WITH A POINTER TO THE LOCATION SYMBOL'S PLACE
0623 (DEFINEE THREAD). THERE ARE ALSO SOME COMPLICATIONS TO COVER THE CASE OF A SYMBOL THAT DEFINES SEVERAL SYMBOLS.
0625 THE UPSHOT OF ALL THIS IS THAT AN INTERMEDIATE PASS, CALLED PASS 1.5, RUNS AROUND THE SYMBOL TABLE CLEANING UP
0627 ALL UNFINISHED EQUALS DEFINITIONS BEFORE PASS 2 BEGINS. IF A DEFINER IS UNDEFINED OR ILL-DEFINED, ITS WHOLE
0629 TREE OF DEFINEES IS CALLED "NEARLY DEFINED," MEANING THAT THE FORM OF THE EQUALS DEFINITION IS ALL RIGHT, AND
0631 ONLY THE DEFINITION OF THE ADDRESS SYMBOL IS LACKING. SIMILARLY, ALL MEMBERS OF A VICIOUS CIRCLE OF EQUALS
0633 DEFINITIONS, AND THEIR OTHER DEFINEES, ARE CALLED "NEARLY DEFINED." TO TAKE SOME EXAMPLES FROM Y/81:

```
0635 OUT1    IS     BNK -4
0636 BNK     EQUALS 15              PUTS INTO THE SYMBOL TABLE:

0637         LOCA:  OUT1                    LOCB:  BNK
0638                -4          (N)               15          (.)     (N FOR NEARLY DEF., = FOR DEF. BY =.)
0640                LOCB        0                 0           LOCA    (DEFINER THREAD AND DEFINEE THREAD)
```

0642 WHEN PASS 1.5 FINDS OUT1 DURING ITS SCAN OF NEARLY DEFINED SYMBOLS, IT FOLLOWS THE DEFINER THREAD TO LOCB, WHERE
0644 IT FINDS THE DEFINITION OF BNK. IT THEN FOLLOWS THE DEFINEE THREAD TO LOCA, AND FILLS IN THE DEFINITION OF
0646 OUT1. THE SYMBOL TABLE ENTRIES BECOME:

```
0647         LOCA:  OUT1                    LOCB:  BNK
0648                11          (=)               15          (.)
0649                0           0                 0           0
```

0650 THE IMPLICATION FOR THE PROGRAMMER IS THAT A TREE OF EQUALS DEFINITIONS MAY HAVE ANY DEPTH, BE OF ANY
0652 COMPLEXITY, AND THE DEFINITIONS MAY BE IN ANY ORDER, PROVIDED ONLY THAT THE TREE IS LOGICALLY VALID. THERE IS,
0654 HOWEVER, ONE VERY SMALL REWARD FOR HAVING A PREDEFINED ADDRESS SYMBOL: THE ADDRESS FIELD MODIFIER MAY LIE
0656 OUTSIDE THE RANGE -32767D TO +32767D, BECAUSE IT DOESN'T HAVE TO FIT INTO A 16-BIT FIELD.

0658 =MINUS

0659 THIS CODE OPERATES BY SUBTRACTING THE LOCATION COUNTER VALUE FROM THE ADDRESS FIELD MODIFIER AND PRO-
0661 CEEDING EXACTLY AS FOR EQUALS, SO THE RESTRICTION AT THE END OF THE PRECEDING NOTE COULD BECOME IMPORTANT.
0663 HOWEVER, IT IT STILL EASY TO GET AROUND: SUPPOSE TABLE AND ENDTABLE ARE KNOWN TO BE IN SUPER-BANK 4, THEN

```
0665          SETLOC TABLE
0666 SIZTABLE =MINUS ENDTABLE        WORKS ONLY IF ENDTABLE IS PREDEFINED.
```

0667 THE ESCAPE IS TO WRITE INSTEAD:

```
0668          SETLOC TABLE -20000
0669 TEMTABLE =MINUS ENDTABLE
0670 SIZTABLE EQUALS TEMTABLE -20000
```

0671 =PLUS

0672 THIS CODE OPERATES BY ADDING THE LOCATION COUNTER VALUE TO THE ADDRESS FIELD MODIFIER AND PROCEEDING
0674 EXACTLY AS FOR EQUALS. THEREFORE REMARKS SIMILAR TO THOSE FOR =MINUS APPLY TO =PLUS, THOUGH THE CASE IS MUCH
0676 LESS LIKELY.

0677 BANK

0678 THIS CODE OPERATES SIMILARLY TO BLOCK, AS DESCRIBED AT Y/77, EXCEPT IN TWO RESPECTS: THE BLOCK SIZE IN
0680 BLK2 AND AGC IS 1024D = 2000, AND THE BANK NUMBERS ARE 4 LESS THAN THE CORRESPONDING BLOCK NUMBERS, BECAUSE BANK
0682 00 STARTS AT PSEUDO-ADDRESS 10000 INSTEAD OF 0. EXAMPLES:

```
0683          BANK   20          FIND FIRST UNUSED LOCATION IN BANK 20
0684          BANK   0 +1400     FIND FIRST UNUSED LOCATION IN LAST PARAGRAPH OF BANK 00
0686          BANK   3           IS ILLEGAL: THAT'S NONEXISTENT MEMORY
0687          BANK               FIND FIRST UNUSED LOCATION IN SAME BANK AS LOCATION COUNTER VALUE
```

0689 SEE ALSO NOTE ON BLOCK.

0690 BBCON*

0691 BITS 7-5 OF A BBCON WORD CONTAIN THE CURRENT SBANK= SETTING (000 IF NONE HAS BEEN ESTABLISHED IN THE
0693 PROGRAM) OR, IF THE FIXED-BANK NUMBER IS 30 OR MORE, IT CONTAINS THE CORRESPONDING SUPER-BANK NUMBER, WHICH IS
0695 ALWAYS THE EIGHTS' DIGIT OF THE BANK NUMBER. IN THE AGC ASSEMBLER, THE E-BANK VALUE MUST BE GIVEN BY A ONE-SHOT
0697 EBANK= DECLARATION. THIS IS OPTIONAL IN BLK2. IN EITHER ASSEMBLER, A ONE-SHOT SBANK= MAY BE USED UNLESS IT
0699 CONFLICTS WITH THE FIXED-BANK VALUE.

0700 THE STARRED VERSION, BBCON*, IS A SPECIAL IMPLIED-ADDRESS CONSTANT CREATED FOR THE BANKSUM CHECKING IN
0702 SELFCHECK. THE ADDRESS FIELD MUST BE LEFT BLANK, AND THE WORD IS ASSEMBLED AS IF THE ADDRESS HAD BEEN THE
0704 HIGHEST ADDRESS OCCUPIED BY A WORD IN THIS ASSEMBLY. ASIDE FROM THAT, ALL THE BBCON RULES APPLY. THE EFFECT OF
0706 THE BBCON* WORD IS TO TELL SELFCHECK WHICH IS THE LAST BANK THAT NEEDS ITS BANKSUM CHECKED.

0708 BLOCK

0709 THE MEMORY BLOCK SIZE IN BLK2 AND AGC IS 1024D = 2000. CONTRARY TO Y77, WHICH GIVES THE BLOCK I AGC
0711 PARAMETERS. THE BLOCK NUMBER IS THEREFORE OBTAINED BY SHIFTING THE PSEUDO-ADDRESS RIGHT 10 BITS, I.E.

0713 BLOCK #: 0 1 2,3 4 5 10 11 ... N
0715 MEMORY: E0-E3 E4-E7 FIXED-FIXED BANK 00 BANK 01 BANK 04 BANK 05 ... BANK N-4

0717 AS FIXED-FIXED MEMORY HAS 2 BLOCK NUMBERS BUT NO BANK NUMBERS, IT IS NECESSARY TO USE BLOCK RATHER THAN BANK
0719 WHEN SEEKING A FREE LOCATION IN FIXED-FIXED.

0720 AN ADDRESS FORMAT HAS BEEN ADDED TO BLOCK AND BANK SINCE Y77 WAS WRITTEN: BLANK OR PLUS-SIGNED NUMERIC:

0722 BLOCK FIND FIRST FREE LOCATION IN SAME BLOCK AS LOCATION COUNTER VALUE
0724 BLOCK +1000 FIND FIRST FREE LOCATION IN LAST HALF OF SAME BLOCK AS LOCATION COUNTER VALUE

0726 THE EFFECT IS TO ALLOW A BLOCK OR BANK CODE TO DEPEND ON A SYMBOLIC ADDRESS FIELD, WHICH IS LOCATED IN A PRE-
0728 CEDING SETLOC.

0729 BNKSUM

0730 THIS CODE GENERATES THE END-OF-BANK PATTERN RECOGNIZED BY SELFCHECK IN ITS CHECKSUM TEST OF FIXED
0732 MEMORY. MORE IMMEDIATELY, THE PATTERN IS RECOGNIZED BY YUL PASS 3, WHICH COMPUTES CHECKSUM WORDS AND MAKES THEM
0734 THE LAST OCCUPIED WORDS IN EACH BANK. THE BLK2 AND AGC ASSEMBLERS TREAT BNKSUM THE SAME IN THE FOLLOWING
0736 RESPECTS:

0737 (1) THE LOCATION FIELD SHOULD BE BLANK.

0738 (2) THE ADDRESS FIELD MUST CONTAIN EITHER A BANK NUMBER (THE USUAL CASE) OR AN ADDRESS IN FIXED MEMORY.
0740 FOR BNKSUM PURPOSES, FIXED-FIXED IS INDICATED BY BANK NUMBERS 2 AND 3.

0742 (3) IF THE INDICATED BANK IS EMPTY, AND NO HIGHER BANK IS OCCUPIED, BNKSUM GENERATES NO WORDS, AND PRINTS
0744 "NO NEED" IN THE ASSEMBLY LISTING. THUS A COMPLETE LIST OF BNKSUM CARDS WILL SERVE A PROGRAM OF VARYING
0746 LENGTH.

0747 (4) BNKSUM TRIES TO GENERATE TWO CONSECUTIVE ADRES-SELF WORDS, THAT IS, TC-TO-SELF WORDS FLAGGED AS CON-
0749 STANTS TO STOP AN ALL-DIGITAL SIMULATION QUICKER THAN A TRUE TC-TO-SELF, IF EXECUTED. THE PLACING OF
0751 THE FIRST WORD IS GOVERNED BY A SEARCH FOR THE FIRST UNOCCUPIED WORD IN THE BANK, REGARDLESS OF WHETHER
0753 THE REST OF THE BANK IS EMPTY. IN PARTICULAR, IF THE FOLLOWING LOCATION IS OCCUPIED, THE SECOND ADRES-
0755 SELF WORD GOES THERE ANYWAY, WITH A CONFLICT CUSS.

0756 IN THE BLK2 ASSEMBLER, AND IN AGC PROGRAM SUNBURST, SELFCHECK LOGIC REQUIRES BOTH ADRES-SELF WORDS
0758 FOLLOWED BY A CHECKSUM WORD, THE LATTER CHOSEN SUCH THAT THE SUM OF ALL WORDS IN THE BANK ADD UP TO THE BANK
0760 NUMBER. THE SUM IS COMPUTED IN A SLIGHTLY SPECIAL WAY, WITH PLUS OR MINUS OVERFLOW BEING CARRIED END-AROUND AS
0762 A PLUS OR MINUS 00001. THE REQUIREMENT FOR BOTH ADRES-SELF WORDS MEANS THAT THE FIRST MAY BE PLACED NO LATER
0764 THAN LOCATION 3775 OF THE BANK. IF ALL WORDS 2000-3775 ARE OCCUPIED, THE ADRES-SELF WORDS GO INTO LOCATIONS
0766 3775 AND 3776 WITH A CONFLICT CUSS.

0767 BNKSUM (CONTINUED)

0768 THE AGC ASSEMBLER RULES CORRESPOND TO A CHANGE IN SELFCHECK THAT WAS MADE AFTER BURST116 WAS FROZEN.
0770 THE CHANGE IS THAT THE ADRES-SELF WORDS ARE PARTLY OR WHOLLY OMITTED IN A NEARLY FULL BANK. THAT IS, IF
0772 LOCATIONS 2000-3775 ARE OCCUPIED, EITHER OR BOTH OF THE ADRES-SELF WORDS ARE OMITTED IN SUCH A WAY THAT THE
0774 CHECKSUM WORD IS IN LOCATION 3777. TO ELIMINATE CONFUSION ABOUT THE AMOUNT OF ROOM LEFT IN A BANK, THE AGC
0776 BNKSUM PROCEDURE PRINTS THE NUMBER OF WORDS (IN DECIMAL) FROM THE LOCATION OF THE FIRST ADRES-SELF WORD THROUGH
0778 LOCATION 3776, THAT IS, THE ABSOLUTE MAXIMUM OF WORDS THAT CAN BE ADDED IN THAT BANK AND STILL LEAVE ONE
0780 LOCATION FOR THE CHECKSUM. THE REQUIREMENT FOR THE CHECKSUM IN THE AGC ASSEMBLER IS DIFFERENT TOO: THE BANK
0782 MUST ADD UP TO PLUS OR MINUS THE BANK NUMBER. TO SEE WHY, IMAGINE THAT BANK 25 IS FULL AND THAT LOCATIONS
0784 2000-3776 ADD UP (CHECKSUM-FASHION) TO 40000, OR -37777. NOW NO CHECKSUM WORD CAN BE CHOSEN THAT WILL YIELD
0786 00025 WHEN ADDED TO 40000, BUT A CHECKSUM WORD OF 37752 YIELDS 77752 = -25. THIS CAN NEVER HAPPEN IN BLK2
0788 BECAUSE THE LAST TWO WORDS BEFORE THE CHECKSUM WORD CONTRIBUTE AT LEAST +4001 TO THE SUM, WHICH THEREFORE HAS A
0790 POSSIBLE RANGE OF -33774 TO +37777. IF BNKSUM FINDS A BANK'S 1024D WORDS ALL OCCUPIED, IT PUTS 03777 INTO
0792 LOCATION 3777 OF THE BANK, WITH A CONFLICT CUSS.

0793 THIS IS A GOOD PLACE TO SAY SOMETHING ABOUT THE COMPUTATION OF THE CHECKSUM WORD IN PASS 3. IT WILL
0795 ACCEPT THE FIRST CONSECUTIVE PAIR OF TC-SELFS (OR ADRES-SELFS) IN A BANK AS THE END-OF-BANK MARKER, AND TRY TO
0797 PUT THE CHECKSUM WORD IN THE FOLLOWING LOCATION, BUT IT'S PERFECTLY HAPPY IF THERE IS NO SUCH PAIR (MEANING NO
0799 CHECKSUM IS WANTED). THE GENERAL PRINCIPLE IS TO GIVE YOU EITHER A CHECKSUM OR A REASON WHY NOT. IF A BANK IS
0801 COMPLETELY FILLED, WITH NO CONSECUTIVE PAIR OF TC-SELFS, THE LAST WORD IN THE BANK GETS THE PREFIX "NOSUM" IN
0803 THE OCTAL STORAGE MAP. IF THERE IS AN UNOCCUPIED LOCATION BEFORE ANY PAIR OF TC-SELFS, IT IS MARKED "NO
0805 CHECKSUM," AND NO ATTEMPT IS MADE TO COMPUTE ONE. IF A LOCATION IS EITHER A "BAD WORD" OR A "CONFLICT," THERE
0807 IS NO CHECKSUM. IF THE LOCATION WHERE THE CHECKSUM WORD SHOULD GO IS OCCUPIED, THE OFFENDING WORD IS PREFIXED
0809 "=SUM" AND THE CHECKSUM IS ABANDONED. IF ANY WORDS AFTER THE CHECKSUM WORD ARE OCCUPIED, THE FIRST IS PREFIXED
0811 "=EOB" (END-OF-BANK ERROR). THE PREFIXES BEGINNING WITH = COUNT AS CUSSED LINES. A PECULIAR CASE ARISES IF THE
0813 TC-SELFS ARE THE LAST TWO WORDS OF A PARAGRAPH BUT NOT THE END OF A BANK, AND THE FOLLOWING PARAGRAPH IS
0815 UNOCCUPIED (BY EXPLICIT CODING, ANYWAY). A PARAGRAPH MUST BE CREATED JUST FOR THE CHECKSUM WORD, AND THE ONLY
0817 FACILITY THAT DOES IT AUTOMATICALLY IS THE BNKSUM ROUTINE IN THE AGC ASSEMBLER.

0819 IN THE AGC COMPUTATION OF THE CHECKSUM WORD, THE RULE IS TO GIVE IT THE SMALLEST MAGNITUDE POSSIBLE.
0821 IF THE SUM FROM THE BEGINNING OF THE BANK THROUGH THE LAST WORD BEFORE THE CHECKSUM IS POSITIVE (NEGATIVE), THE
0823 CHECKSUM WORD IS CHOSEN TO MAKE THE WHOLE BANK ADD UP TO PLUS (MINUS) THE BANK NUMBER. IN AGC OR BLK2, BY THE
0825 WAY, BANK 00 HAS TO ADD UP TO 77777 (-0).

0826 EBANK=

0827 THE AGC ASSEMBLER REQUIRES THE ADDRESS VALUE FOR EBANK= TO BE AT LEAST 10 AND AT MOST 3777--THAT IS, AN
0829 ADDRESS IN ERASABLE MEMORY. THE BLK2 ASSEMBLER ALLOWS THESE, AND ALSO ALLOWS E-BANK NUMBERS 0-7.

0831 THE AGC ASSEMBLER REQUIRES A ONE-SHOT EBANK= DECLARATION BEFORE EVERY BBCON OR BBCON* AND 2BCADR OR
0833 2CADR; THE BLK2 ASSEMBLER DOES NOT.

0834 EDRUPT

0835 THE BLK2 ASSEMBLER REGARDS EDRUPT AS AN IMPLIED-ADDRESS EXTRACODE AND ALWAYS ASSEMBLES IT 07000. THE
0837 AGC ASSEMBLER REGARDS IT AS A FULL EXTRACODE REQUIRING A FIXED-MEMORY ADDRESS, AND IT IS THE AGC PROGRAMMER'S
0839 RESPONSIBILITY TO SEE THAT THE ADDRESS IS BETWEEN 7000 AND 7777. THIS IS BECAUSE, IN THE FLIGHT-ROPE APPLICA-
0841 TION, CONTROL RETURNS (AFTER CERTAIN RESUMES) TO THE EDRUPT INSTRUCTION, NOT TO THE EXTEND BEFORE IT, SO THAT IT
0843 IS EXECUTED AS A TC INSTRUCTION. SEE T5JOB IN SUNDANCE FOR FURTHER INFORMATION.

0845 EVEN

0846 THIS CODE TAKES NO LOCATION OR ADDRESS FIELDS. IF THE LOCATION COUNTER VALUE IS EVEN, IT HAS NO EFFECT.
0848 IF THE LOCATION COUNTER VALUE IS ODD, IT IS INCREMENTED. THIS IS USEFUL IF IT'S DESIRED TO ALIGN DOUBLE-WORD
0850 QUANTITIES ON EVEN LOCATIONS, WHICH INCIDENTALLY GUARDS AGAINST BANK-STRADDLING. HOWEVER, IT SHOULD BE USED IN
0852 CONJUNCTION WITH LEFTOVER CONSTANTS TO FILL UP THE RESULTING HOLE. DON'T FEEL BAD IF YOU CAN'T USE IT.

0854 MM

0855 THIS CODE IS EQUIVALENT TO DEC EXCEPT THAT THE ARGUMENT IS LIMITED TO A MAGNITUDE OF 99. IT WAS INTRO-
0857 DUCED TO EASE THE TRANSITION FROM OCTAL TO DECIMAL MAJOR MODES, AND NOW HAS MOSTLY EYEBALL VALUE.

0859 MODIFY

0860 THIS CODE IS LISTED HERE MOSTLY FOR THE BENEFIT OF THOSE (LIKE THE YUL-360 PEOPLE) WHO READ THE YUL-
0862 1800 PROGRAM. THE CODE IS RECOGNIZED BUT IT IS NOT NOW INTENDED TO BE USED, AT LEAST IN FLIGHT PROGRAMS. IT
0864 TAKES NO LOCATION OR ADDRESS FIELD, AND ITS FUNCTION IS TO ACCEPT ALL CARD IMAGES COMING IN FROM TAPE IN A
0866 REVISION OR VERSION ASSEMBLY OR FROM THE DISC IN A TRANSFERRED ASSEMBLY, UP TO AND INCLUDING THE NEXT "END OF"
0868 CARD IMAGE, WHICH MARKS THE END OF EVERY PROGRAM AND SUBROUTINE FILE ON TAPE. THEN FURTHER CHANGE CARDS FROM
0870 THE CARD READER ARE MERGED WITH A SUBROUTINE CALLED BY THE MAIN PROGRAM, FOR THE DURATION OF THE CURRENT ASSEM-
0872 BLY ONLY. THE SUBROUTINE FILES ON TAPE ARE UNCHANGED. THE YUL-360 PEOPLE ARE BEGINNING TO USE THE WORD
0874 "MODIFY" IN THE SENSE OF "REVISE A SUBROUTINE."

0875 SUBRO*

0876 THIS CODE MAY ONLY BE USED IN A PROGRAM, I.E. SUBROUTINES CAN'T CALL OTHER SUBROUTINES. THE SUPPRESSION
0878 OF PRINTING CALLED FOR BY THE SUBDIRECTOR "SUPPRESS INACTIVE SUBROUTINES" WORKS THIS WAY: IF A SUBROUTINE HAS
0880 NOT BEEN CHANGED SINCE THE LAST ASSEMBLY OF THE PROGRAM CALLING IT, IT IS "INACTIVE," AND IF THE SUPPRESSION IS
0882 ON, THE ONLY LINES PRINTED ARE THE "END OF" LINE AND ANY CUSSED LINE WITH ITS CUSSES. THESE ARE MADE TO APPEAR
0884 UNDER THEIR PROPER LOG SECTIONS.

0885 A SUBROUTINE MAY BE MADE PERMANENTLY ACTIVE WITH RESPECT TO A GIVEN PROGRAM IF IT IS CALLED BY SUBRO*
0887 INSTEAD OF SUBRO. ALL SUBROUTINES CALLED IN A NEW, VERSION, OR TRANSFERRED PROGRAM ASSEMBLY ARE NECESSARILY
0889 ACTIVE, SINCE A NEW PROGRAM IS BEING CREATED.

0890 VN, <NV>

0891 THIS IS A VARIATION ON DEC TO PACK TWO SEVEN-BIT FIELDS, REPRESENTING A VERB AND A NOUN, INTO ONE AGC
0893 WORD. THE VERB AND NOUN NUMBERS ARE WRITTEN IN THE ADDRESS FIELD IN DECIMAL, IN THE FORM VVNN (OR INDEED ANY
0895 FORM ACCEPTABLE TO DEC THAT COMES TO THE SAME VALUE). THE MAGNITUDE OF THIS EXPRESSION IS LIMITED TO 9999.
0897 THE PARTS VV AND NN ARE IN EFFECT CONVERTED TO BINARY SEPARATELY AND PLACED IN BITS 14-8 AND 7-1 OF THE WORD,
0899 RESPECTIVELY.