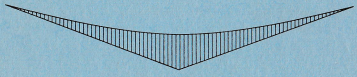
Geo. L. Silver DINO

APOLLO GUIDANCE COMPUTER

PROGRAMMING INFORMATION



SYSTEMS SUPPORT TRAINING

RAYTHEON

RAYTHEON COMPANY
SPACE AND INFORMATION SYSTEMS DIVISION

CONTENTS

BASIC INSTRUCTIONS

ADS E 02.6	ADS0 STD2	"Add to Storage E" Adds c(A) and c(E), stores sum with overflow bit in A and sum without overflow bit in E.
AD K 06.	AD0 STD2	"Add K" Adds c(K) to c(A) and stores sum in A.
AUG E 12.4	AUG0 STD2	"Augment E" Increases the magnitude of the quantity contained in E by one and stores the augmented quantity in E.
BZF F 11.2 11.4 11.6	BZF0 STD2	"Branch on Zero to Fixed F" Branches according to c(A). c(A) — Transfers to plus or minus zero — F (subinstruction STD2 is not executed) non-zero — I+1 (subinstruction STD2 is executed)
BZMF F 16.2 .4 .6	BZMF0 STD2	"Branch on Zero or Minus to Fixed F" Branches according to c(A). c(A) — Transfers to zero or negative non-zero — F (subinstruction STD2 is not executed) positive non-zero — I+1 (subinstruction STD2 is executed)
CCS E 01.0	CCS0 STD2	"Count, Compare, and Skip on E" Branches according to c(E) and stores in A the | c(E) | diminished by one. c(E) — Transfers to positive non-zero — I+1 plus zero — I+2 negative non-zero — I+3 minus zero — I+4
CA K 03.	CA0 STD2	"Clear and Add K" Enters c(K) into A
CYL .0022		"Cycle Left" Cycles quantity, which is entered into location 0022, one place to the left.
CS K 04.	CS0 STD2	"Clear and Subtract K" Enters the complemented c(K) into A.
CYR .0020		"Cycle Right" Cycles quantity, which is entered into location 0020, one place to the right.
DAS E (E+1)	DAS0 DAS1 STD2	"Double Add to Storage E" Adds c(A, L) and c(E, E+1) and stores sum without overflow bit in E and E+1. Enters plus one into A in case of positive overflow, minus one in case of negative overflow, and plus zero in case of no overflow. Enters plus zero into L.
DCA K (K+1) 13.	DCA0 DCA1 STD2	"Double Clear and Add K" Enters c(K, K+1) into A and L.
DCS K (K+1) 14.	DCS0 DCS1 STD2	"Double Clear and Subtract K" Enters the complemented c(K, K+1) into A and L.
DIM E 12.6	DIM0 STD2	"Diminish E" Decreases the magnitude of the quantity contained in E by one and stores diminished quantity in E.
DINC C	DINC	"Diminish Increment C" Decreases the magnitude of the quantity contained in C by one and stores diminished quantity in C.

DV E 11.0	DV0 DV1 DV3 DV7 DV6 DV4 STD2	"Divide by E" Divides double precision quantity c(A, L) by c(E), stores quotient in A and remainder in L. Signs of b(A) and b(L) need not agree. Sign of remainder equals sign of dividend.
DXCH E (E+1) 05.2	DXCH0 DXCH1 STD2	"Double Exchange A and E" Exchanges c(A, L) with c(E, E+1).
EDOP .0023		"Edit Operator" Shifts quantity, which is entered into location 0023, seven places to the right.
EXTEND 00.0006	STD2	"Extend" Enters a ONE into bit position SQ-EXT. The next instruction, taken from I+1, is an Extra-Code Instruction
FETCH K	FETCH0 FETCH1	"Fetch K" Displays c(K) on GSE. Address K is supplied by GSE.
GO 00.	GOJ1 TC0	"Go" Takes next instruction from location 4000 in F Memory.
INCR E 02.4	INCR0 STD2	"Increment E" Adds plus one to c(E) and stores incremented quantity in E.
INHINT 00.0004	STD2	"Inhibit Interrupt" Sets inhibit interrupt switch in Interrupt Priority Control to prevent interruption of program execution.
INOTLD H	INOTLD	"In Out Load H" Data supplied by GSE is entered into H by GSE. Channel address H is supplied by GSE.
INOTRD H	INOTRD	"In Out Read H" Displays c(H) on GSE. Channel address H is supplied by GSE.
LXCH E 02.2	LXCH0 STD2	"Exchange L and E" Exchanges c(L) with c(E).
MCDU C	MCDU	"Minus CDU C" Subtracts one from cyclic TWO's complement number in C and stores decremented quantity in C.
MINC C	MINC	"Minus Increment C" Subtracts one from c(C) and stores decremented quantity in C.
MP K 17.	MP0 MP1 MP3	"Multiply K" Multiplies c(K) by c(A) and stores double precision product in A and L (signs in A and L agree).
MSK K 07.	MSK0 STD2	"Mask with K" AND's c(A) with c(K) and stores logical product in A.
MSU E 12.0	MSU0 STD2	"Modular Subtract E" Subtracts cyclic TWO's complement number in E from cyclic TWO's complement number in A and stores difference expressed in ONE's complement number in A.
NDX E 05.0	NDX0 NDX1	"Index Next Basic Instruction with E" Adds c(E) to c(I+1) and takes sum as next instruction.
NDX K 15.	NDXX0 NDXX1	"Index Next Extra-Code Instruction with K" Adds c(K) to c(I+1) and takes sum as next instruction. Retains the ONE in bit position SQ-EXT.
PCDU C	PCDU	"Plus CDU C" Adds one to cyclic TWO's complement number in C and stores incremented quantity in C.

PINC C	PINC	"Plus Increment C" Adds one to c(C) and stores incremented quantity in C.
QXCH E 12.2	QXCH0 STD2	"Exchange Q and E" Exchanges c(Q) with c(E). Takes next instruction from I+1.
RAND H 10.2	RAND0 STD2	"Read and AND H" ANDs c(A) and c(H) and stores logical product in A.
READ H 10.0	READ0 STD2	"Read H" Enters c(H) into A.
RELINT 00.0003	STD2	"Release Inhibit Interrupt" Resets inhibit interrupt switch to allow program interruption in favor of a programmed operation of higher priority.
RESUME 05.0017	NDX0 RSM3	"Resume Interrupted Program" Takes next instruction from location 0017 and enters content of location 0015 into Z. Thus, execution of the interrupted program section is resumed.
ROR H 10.4	ROR0 STD2	"Read and OR H" ORs c(A) and c(H), and stores logical sum in A.
RUPT 10.7	RUPT0 RUPT1 STD2	"Interrupt Program Execution" Takes next instruction from address supplied by Interrupt Priority Control. Stores c(B) in location 0017 and c(Z) in location 0015.
RXOR H 10.6	RXOR0 STD2	"Read and Exclusive OR H" Forms exclusive OR from c(A) and c(H), and stores result in A.
SHANC C	SHANC	"Shift and Add Increment C" Shifts c(C) one place to the left and enters a ONE into bit position 1 of C.
SHINC C	SHINC	"Shift Increment C" Shifts c(C) one place to the left and enters a ZERO into bit position 1 of C.
SR .0021		"Shift Right" Shifts quantity, which is entered into location 0021, one place to the right.
STORE E	STORE0 STORE1	"Store E"; data supplied by GSE is entered into E by GSE. Address E is also supplied by GSE.
SU E 16.0	SU0 STD2	"Subtract E" Subtracts c(E) from c(A) and stores the difference in A.
TC K 00.	TC0	"Transfer Control to K" Takes next instruction from K and stores return address (I+1) in Q.
TCF F 01.2 .4 .6	TCF0	"Transfer Control to Fixed F" Takes next instruction from F without changing c(Q).
TCSAJ K 00.	TCSAJ3 STD2	"Transfer control to specified address K" Takes next instruction from address which is supplied by GSE.

TS E 05.4	TS0 STD2	"Transfer to Storage E" If A does not contain an overflow quantity, instruction enters c(A) into E and takes next instruction from I+1. If A contains a positive overflow, instruction enters c(A) without overflow bit into E, enters plus one into A, and takes next instruction from I+2. If A contains a negative overflow, instruction enters c(A) without overflow bit into E, enters minus one into A, and takes next instruction from I+2.
WAND H 10.3	WAND0 STD2	"Write and AND H" ANDs c(A) and c(H), and stores logical product in A and H.
WOR H 10.5	WOR0 STD2	"Write and OR H" ORs c(A) and c(H), and stores logical sum in A and H.
WRITE H 10.1	WRITE0 STD2	"Write H" Enters c(A) into H.
XCH E 05.6	XCH0 STD2	"Exchanges A and E" Exchanges c(A) with c(E).

INTERPRETIVE INSTRUCTIONS

ABS 0	TP Absolute Value (0.48 ms)	|T(MPAC)| replace T(MPAC).
ABVAL 0	Vector Length (3.86 ms)	|V(MPAC)| becomes T(MPAC), changing the store mode to DP. In addition, $\|V(MPAC)\|^2$ replace D(34D). The result is zero if $\|V(MPAC)\|^2 \geq 2^{-21}$. If $\|V(MPAC)\| > 1$ set OVFIND to indicate unspecified result.
ARCCOS (ACOS) 0	DP Arc-Cosine (9.12 ms)	$(1/2\pi)$ Arc-Cosine (ZD(MPAC) replace T(MPAC). This is the inverse of COS. Receipt of an argument whose magnitude is greater than .5001 causes an abort.
ARCSIN (ASIN) 0	DP Arc-sine (9.26 ms)	$(1/2\pi)$ Arc-sine (ZD(MPAC) replace T(MPAC). This is the inverse of the SIN function. Receipt of an argument greater than .5001 in magnitude causes an abort.
AXC, 1 AXC, 2 2	X Address to X Index Complemented (0.76 ms)	-X replaces S(XT).
AXT, 1 AXT, 2 2	X Address to X Index True (0.75 ms)	X replaces S(XT) (T = 1, 2).
BDDV 1 3	X DP Divide Into (2.50 ms)	Same as DDV except Q=D(X)/D(MPAC) if $\|D(X)\| < \|D(MPAC)\|$.
BDSU 1 3	X DP Subtract From (0.74 ms)	D(X) - D(MPAC) replace D(MPAC). Set OVFIND on overflow, and overflow-correct the result.
BHIZ 2	X Branch High (0.6 ms) Order Zero (+0.19 ms) GO	If S(MPAC) = 0, do a GOTO X. Otherwise, no operation occurs.
BMN 2	X Branch Minus (0.67 ms) (+0.19 ms) GO	If T(MPAC) < 0, do a GOTO X. Otherwise, no operation occurs.
BOFCLR 2	X Branch if Switch (1.36 ms) Y Off, Clearing Switch (+0.23 ms) GO	Clear switch X to 0. If initially cleared to 0, do a GOTO Y. Otherwise, no further operation occurs.
BOFF 2	X Branch if (1.27 ms) Y Switch Off (+0.23 ms) GO	If switch X is cleared to 0, do a GOTO Y. Otherwise, no operation occurs.
BOFINV 2	X Branch if Switch (1.39 ms) Y Off, Inverting Switch (+0.23 ms) GO	Invert switch X. If originally cleared to 0, do a GOTO Y. Otherwise, no operation occurs.
BOFSET 2	X Branch if Switch (1.39 ms) Y Off, Setting Switch (+0.23 ms) GO	Set switch X to 1. If initially cleared to 0, do a GOTO Y. Otherwise, no further operation occurs.
BON 2	X Branch if (1.26 ms) Y Switch On (+0.23 ms)	If switch X is set to 1, do a GOTO Y. Otherwise, no further operation occurs.
BONCLR 2	X Branch if Switch (1.35 ms) Y On, Clearing Switch (+0.23 ms) GO	Clear switch X to 0. If initially set to 1, do a GOTO Y. Otherwise, no further operation occurs.

BONINV 2	X Branch if Switch (1.37 ms) Y On, Inverting Switch (+0.23 ms) GO	Invert switch X. If originally set to 1, do a GOTO Y. Otherwise, no further operation occurs.
BONSET 2	X Branch if Switch (1.37 ms) Y On, Setting Switch (+0.23 ms) GO	Set switch X to 1. If initially set to 1, do a GOTO Y. Otherwise, no further operation occurs.
BOV 2	X Branch On (0.58 ms) Overflow (+0.23 ms) GO	If OVFIND is set, reset it to zero and do a GOTO X. Otherwise no operation occurs.
BOVB 2	X Branch On (0.58 ms) Overflow to Basic (+0.16 ms) GO	If OVFIND is set, reset it to zero and begin executing basic instructions at X. Otherwise, no operation occurs. X must be in fixed memory.
BPL 2	X Branch Plugs (0.65 ms) (+0.19 ms) GO	If T(MPAC) > 0, do a GOTO X. Otherwise, no operation occurs.
BVSU 1 3	X Vector Subtract From (1.17 ms)	V(X) - V(MPAC) replace V(MPAC). Set OVFIND on overflow of any component, leaving an overflow-corrected result.
BZE 2	X Branch Zero (0.65 ms) (+0.19 ms) GO	If T(MPAC) = 0, do a GOTO X. Otherwise, no operation occurs.
CALL 2	X Call a Subroutine (0.89 ms)	Begin executing interpretive instructions at X. A return address is left in QPRET. Call is a right-hand operation code.
CCALL 2	X Computed Y Call (1.07 ms)	Same as CGOTO except that a return address is left in QPRET in addition. CCALL is a right-hand op code.
CLEAR 2	X Clear Switch (1.25 ms)	Clear switch X to 0.
CLRGO 2	X Clear Switch Y and Go To (1.52 ms)	Clear switch X to 0 and do a GOTO Y. CLRGO is a right-hand op code.
CGOTO 2	X Computer Y Go To (0.90 ms)	The contents of X (X in erasable) are added to address Y (Y in fixed) and the address at Y+S(X) is selected. Begin executing interpretive instructions there unless the address is in erasable, in which case it is interpreted as indirect. CGOTO is a right-hand op code.
CCS (COSINE) 0	DP Cosine (5.80 ms)	5 (Cos (2π D(MPAC)) replace T(MPAC).
DAD 1 3	X DP Add (0.66 ms)	D(MPAC) + D(X) replace D(MPAC). Set OVFIND on overflow, and leaves the overflow-corrected result in MPAC.
DCOMP 0	TP Complement (0.52 ms)	-T(MPAC) replace T(MPAC).
DDV 1 3	X DP Divide By (2.48 ms)	If |D(MPAC)| < |D(X)|, the DP quotient Q= D(MPAC) / D(X) is formed and (Q, 0) replace T(MPAC). Overflow indication is set if required. ± .999999 replace D(MPAC) in this case.
DLOAD 1 3	X Load MPAC in DP (0.64 ms)	(D(X), 0) become T(MPAC), setting the store mode to DP. Address may be direct, indexed or vacuous.

DMP 1 3	X DP Multiply (1.13 ms)	D(X) times D(MPAC) replace T(MPAC).
DMPR 1 3	X DP Multiply and Round (1.29 ms)	D(MPAC) D(X) = P is formed and rounded to DP so that (P, 0) replace T(MPAC).
DOT 1 3	X Vector Dot Product (3.08 ms)	V(MPAC) · V(X) replace T(MPAC), setting the store mode to DP. Set OVFIND if overflow occurs, leaving an overflow-corrected result.
DSQ 0	DP Square (0.76 ms)	D(MPAC) times D(MPAC) replace T(MPAC).
DSU 1 3	X DP Subtract (0.66 ms)	D(MPAC) -D(X) replace D(MPAC). Set OVFIND on overflow, and overflow-correct the result.
EXIT 2	Exit from Interpreter (0.26 ms)	Begin executing basic instructions after the last op code or address word referenced by the interpreter as follows: 1) If EXIT is a left-hand op code, go to the word after the EXIT instructions; 2) If EXIT is a right-hand op code, go to the word following the last address used by the left-hand op code. EXIT is a right-hand op code.
GOTO 2	X Go To (0.77 ms)	Begin executing interpretive instructions at X. QPRET is undisturbed. GOTO is a right-hand operation code.
INCR, 1 INCR, 2 2	X Increment Index X (0.76 ms)	The overflow-corrected sum of S(XT) and X replaces S(XT).
INVERT 2	X Invert Switch (1.27 ms)	Invert switch X; i.e., if 0, set to 1; if 1, clear to 0.
INVGO 2	X Invert Switch Y and Go To (1.54 ms)	Invert switch X and do a GOTO Y. INVGO is a right-hand op code.
LXA, 1 LXA, 2 2	X Load Index X from Erasable (0.78 ms)	S(X) replaces S(XT).
LXC, 1 LXC, 2 2	X Load Index X from Erasable Complemented (0.78 ms)	-S(X) replaces S(XT).
MXV 1 3	X Matrix Post-Multiplication by Vector (8.97 ms)	M(X) V(MPAC) replace V(MPAC). Set OVFIND on overflow, leaving an overflow-corrected result.
NORM (SLC) 1 3	X Scaler Normalize (0.88 ms) (+0.21 N ms)	An N is found such that $\|T(MPAC)\| 2^N \geq .5$ provided T(MPAC) ≠ 0. -N replaces S(X) and T(MPAC) x 2^N replace T(MPAC). If T(MPAC) = 0, -0 replaces S(X) and T(MPAC) are unchanged.
PDDL 1 3	X Push Down and load MPAC in DP (0.91 ms)	D(MPAC), T(MPAC) or V(MPAC) are pushed down: (D(X), 0) become T(MPAC) with the store mode set to DP. X may be direct, indexed, or vacuous.

PDVL 1 3	X Push Down and load MPAC with a vector (1.14 ms)	Same as PDDL except V(X) becomes V(MPAC) and the store mode is set to vector.
PUSH 1 3	Push Down (0.55 ms)	D(MPAC), T(MPAC) or V(MPAC) are pushed down.
ROUND 0	Round to DP (0.56 ms)	T(MPAC) are rounded to DP so that (ROUND (T(MPAC)), 0) replace T(MPAC). Set OVFIND if overflow occurs, leaving an overflow-corrected result, ±0.
RTB 2	X Return to Basic (0.71 ms)	Begin executing basic instructions at X. X must be in fixed memory.
RVQ (ITCQ) 2	Return Via QPRET (0.69 ms)	Begin executing interpretive instructions at the location whose address is in QPRET. This may be used to return from a subroutine which contains no CALL or CCALL instructions. If QPRET contains the address of an erasable register, the address is interpreted as an indirect address. RVQ is a 'right-hand op code'.
SET 2	X Set Switch (1.27 ms)	Set switch X to 1.
SETGO 2	X Set Switch Y and Go To (1.54 ms)	Set switch X to 1 and do a GOTO Y. SETGO is a right-hand op code.
SETPD 2	X Set Push-down Pointer (0.58 ms)	Set the Push-down Pointer PUSHLOC to X, where X is in local erasable memory. X must be direct.
SIGN 1 3	X DP Sign Test (0.70 ms)	X must be in erasable memory. If D(X) ≥ 0, no operation occurs. Otherwise if store mode is DP or TP, -T(MPAC) replace T(MPAC); if store mode is vector, -V(MPAC) replace V(MPAC).
SIN (SINE) 0	DP Sine (5.63 ms)	.5 (Sin (2π D(MPAC)) replace T(MPAC).
SL 1 3	X General Scaler Shift (1.03 ms) Left (+0.22 X ms)	Same as SR except that T(MPAC)2^X replace T(MPAC).
SL1 SL2 SL3 SL4 0	Scaler Shift Left (0.72 ms) (0.92 ms) (1.17 ms) (1.39 ms)	T(MPAC) x 2^j replace T(MPAC) (j = 1, 2, 3, 4). If significant bits are lost, set OVFIND but leave the overflow-corrected result as T(MPAC).
SLOAD 1 3	X Load MPAC in Single Precision (0.74 ms)	Same as DLOAD except (S(X), 0, 0) becomes T(MPAC). X may not be vacuous.
SLR 1 3	X General Scaler Shift (1.18 ms) Left and Round (+0.22 X ms)	Same as SL except that T(MPAC) x 2^X is rounded to a DP number R and (R, 0) replace T(MPAC). Direct address limits are 0 < X < 14.
SL1R SL2R SL3R SL4R 0	Scaler Shift Left and Round (0.86 ms) (1.10 ms) (1.32 ms) (1.54 ms)	T(MPAC) x 2^j is rounded to a DP number R and (R, 0) replace T(MPAC) (j = 1, 2, 3, 4). If overflow occurs, set OVFIND and leave the overflow-corrected result as T(MPAC).

SR 1 3	X General Scaler Shift (1.38 ms) Right (+0.23 INTEGER (X/14) ms)	T(MPAC) x 2^{-X} replace T(MPAC) where $-42 < X < 42$ (X can be negative only if the address was indexed). Address limits are $0 < X < 42$ if direct and $-128 < X_8 < 128$ if indexed. X_8 is the stored address before index modification. X is the net address in any case. On overflow leave the overflow-corrected result and set OVFIND.
SR1 SR2 SR3 SR4 0	Scaler Shift Right (0.85 ms) (0.85 ms) (0.85 ms) (0.85 ms)	T(MPAC) 2^{-j} replace T(MPAC) (j = 1, 2, 3, 4).
SR1R SR2R SR3R SR4R 0	Scaler Shift Right and Round (0.99 ms) (0.99 ms) (0.99 ms) (0.99 ms)	T(MPAC) x 2^{-j} is rounded to a DP number R and (R, 0) replace T(MPAC) (j = 1, 2, 3, 4).
SRR 1 3	X General Scaler Shift (1.52 ms) Right and Round (+0.23 INTEGER (X/14) ms	Same as SR except that T(MPAC) x 2^{-X} is rounded to a DP number R and (R, 0) replace T(MPAC). Address limits are $0 < X < 29$ if direct.
SSP 2	X Set Single Y Precision (0.67 ms)	Y replaces S(X). Y may be any constant: arithmetic, logical, address, etc.
SQRT 0	DP Square Root (1.94 ms)	SQRT (D(MPAC)) replace T(MPAC): i.e. the initial contents of MPAC are normalized, the DP square root of the normalized number computed, and that result unnormalized so that MPAC +2 has marginal significance. Receipt of an argument less than -10^{-4} causes an abort.
STQ (ITAL) 2	X Store QPRET (0.69 ms)	S(QPRET) replaces S(X) (X in erasable). This may be used to save the return address in subroutines which contain CALL and CCALL instructions. The STQ X in this case is eventually followed by GOTO X to return.
STADR 0	Push Up On Store Code (0.26 ms)	During assembly, the appearance of STADR causes the next store code to be stored complemented. During execution, STADR complements the next word to be referenced by the interpreter and enters the store code processor. STADR is a right-hand op code.
STCALL 1 3	X Store MPAC Y and CALL a Routine (1.40 ms)	D(MPAC), T(MPAC), or V(MPAC) replace D(X), T(X) or V(X), leaving the store mode unaltered. Call the routine at Y, leaving a return address (of the location after the second address) in QPRET. Both addresses must be direct.
STODL 1 3	X Store MPAC Y and re-load in DP (1.24 ms)	D(MPAC), T(MPAC) or V(MPAC) replace D(X), T(X) or V(X). (D(Y), 0) becomes T(MPAC) setting the store mode to DP. X may be indexed, or direct and Y indexed, direct or vacuous (push-up).
STORE 1 3	X Store MPAC (0.62 ms)	D(MPAC), T(MPAC) or V(MPAC) replace D(X), T(X), or V(X), respectively. X may be indexed or direct.
STOVL 1 3	X Store MPAC Y and re-load as Vector (1.43 ms)	Same as STODL except V(X) become V(MPAC) and store mode is set to vector.
SXA, 1 SXA, 2 2	X Store Index X in Erasable (0.78 ms)	S(XT) replaces S(X).

TAD	X TP Add (0.75 ms)	T(MPAC) + T(X) replace T(MPAC). OVFIND is set on overflow, with the overflow-corrected result left in MPAC.
TIX, 1 TIX, 2 2	X Transfer on Index (0.78 ms) X (+0.26 ms) GO	If S(XT) ≤ S(ST) (T = 1, 2), no operation occurs. Otherwise, S(XT) − S(ST) replaces S(XT) and a GOTO X is executed.
TLOAD 1 3	X Load MPAC in TP (0.77 ms)	Same as DLOAD except T(X) becomes T(MPAC) and store mode is set to TP.
UNIT 0	Unit Vector Function (6.46 ms)	$V(MPAC)/\frac{2}{c}\|V(MPAC)\|$ replace V(MPAC). $\|V(MPAC)\|^2$ replace D(34D) and $\|V(MPAC)\|$ replace D(36D). Set OVFIND if $\|V(MPAC)\| < 2^{-21}$ or $\|V(MPAC)\|$ in which case the result is incorrect.
VAD 1 3	X Vector Add (0.92 ms)	V(MPAC + V(X) replace V(MPAC). Set OVFIND on overflow in any component, leaving the overflow-corrected result.
VCOMP 0	Vector Complement (0.63 ms)	−V(MPAC) replace V(MPAC).
VDEF 0	Vector Define (0.67 ms)	Push up for V_Y and again for V_Z so that (D(MPAC), V_X, V_Y, V_Z) become V(MPAC), setting the store mode to vector.
VLOAD 1 3	X Load MPAC with a Vector (0.91 ms)	Same as DLOAD except V(X) becomes V(MPAC) and store mode is set to vector.
VPROJ 1 3	X Vector Projection (3.75 ms)	$[V(MPAC) \cdot V(X)]$ V(X) replace V(MPAC). Set OVFIND on overflow, and leave the result obtained with overflow-corrected $[V(MPAC) \cdot V(X)]$
V/SC 1 3	X Vector Divided by Scaler (5.39 ms)	If the initial store mode is Vector, each component of V(MPAC) is divided by D(X), the DP quotients replacing their respective components of V(MPAC). If the initial store mode is DP or TP, it is changed to Vector, and each component of V(X) is divided by D(MPAC) to form V(MPAC). If overflow occurs in any component, the operation is terminated with OVFIND set and unspecified results in MPAC.
VSL 1 3	X General Vector Shift (0.89 ms) Left (+0.37 X ms)	Each component of V(MPAC) is replaced by the original component multiplied by 2^X. On overflow of any component, leave the overflow-corrected result and set OVFIND. If the address was indexed and the resulting address negative, VSR(−X) instead. Address limits are $0 < X < 28$ if direct.
VSL1 VSL2 VSL3 VSL4 VSL5 VSL6 VSL7 VSL8 0	Vector Shift Left (0.81 ms) (1.18 ms) (1.55 ms) (1.93 ms) (2.30 ms) (2.68 ms) (3.05 ms) (3.43 ms)	Each component of V(MPAC) is replaced by the original value multiplied by 2^{-j} (j = 1 (1) 8). If overflow occurs in any component, leave the overflow-corrected result and set OVFIND.
VSQ 0	Square of Vector Length (2.21 ms)	$\|V(MPAC)\|^2$ becomes T(MPAC), changing the store mode to DP. If $\|V(MPAC)\| \geq 1$, set OVFIND and leave an overflow-corrected result.

VSR 1 3	X General Vector Shift (2.61 ms) Right (+0.82 INTEGER (X/14) ms)	Each component of V(MPAC) is replaced by the original value multiplied by 2^{-X} and rounded to DP. If X is an indexed address and the result address negative, do a VSL -X instead. Address limits are $0 < X < 29$ if direct and $-128 < X_8 < 128$ is indexed.
VSR1 VSR2 VSR3 VSR4 VSR5 VSR6 VSR7 VSR8 0	Vector Shift Right and Round (2.01 ms) (2.01 ms) (2.01 ms) (2.01 ms) (2.01 ms) (2.01 ms) (2.01 ms) (2.01 ms)	Each component of V(MPAC) is replaced by the original value multiplied by 2^{-j} and rounded to DP. (j = 1(1)8).
VSU 1 3	X Vector Subtract (0.92 ms)	V(MPAC) - V(X) replace V(MPAC). Set OVFIND on overflow in any component, leaving an overflow-corrected result.
VXM 1 3	X Matrix Pre- Multiplication by Vector (8.98 ms)	$(V(MPAC)^T M (X))^T$ replace V(MPAC). Set OVFIND on overflow, leaving an overflow-corrected result.
VXSC 1 3	X Vector Times Scaler (3.27 ms)	If the initial store mode is Vector, each component of V(MPAC) is multiplied by D(X), the rounded products replacing their respective X components of V(MPAC). If the initial store mode is DP or TP, change it to Vector, and each component of V(X) is multiplied by D(MPAC) to form V(MPAC) as above.
VXV 1 3	X Vector Cross Product (4.98 ms)	V(MPAC) - V(X) replace V(MPAC). Set OVFIND if overflow occurs, leaving an overflow-corrected result.
XAD, 1 XAD, 2 2	X Index Register X Add (0.77 ms)	The overflow-corrected sum of S(XT) and S(X) replace S(XT).
XCHX, 1 XCHX, 2 2	X Exchange Index X with Erasable (0.83 ms)	S(XT) replaces S(X) which then replaces S(XT).
XSU, 1 XSU, 2 2	X Index Register Subtract X (0.78 ms)	The overflow-corrected difference S(XT) - S(X) replaces S(XT).

VERBS AND NOUNS

REGULAR VERBS
(Command Module)

01	DISPLAY OCTAL COMP 1 (R1)
02	DISPLAY OCTAL COMP 2 (R1)
03	DISPLAY OCTAL COMP 3 (R1)
04	DISPLAY OCTAL COMP 1, 2 (R1, R2)
05	DISPLAY OCTAL COMP 1, 2, 3 (R1, R2, R3)
06	DECIMAL DISPLAY
07	DP DECIMAL DISPLAY (R1, R2)
10	SPARE
11	MONITOR OCT COMP 1 (R1)
12	MONITOR OCT COMP 2 (R1)
13	MONITOR OCT COMP 3 (R1)
14	MONITOR OCT COMP 1, 2 (R1)
15	MONITOR OCT COMP 1, 2, 3 (R1, R2, R3)
16	MONITOR DECIMAL
17	MONITOR DP DECIMAL (R1, R2)
20	SPARE
21	LOAD COMP 1 (R1)
22	LOAD COMP 2 (R2)
23	LOAD COMP 3 (R3)
24	LOAD COMP 1, 2 (R1, R2)
25	LOAD COMP 1, 2, 3 (R1, R2, R3)
26	SPARE
27	FIXED MEMORY DISPLAY
30	REQUEST EXECUTIVE
31	REQUEST WAITLIST
32	C(R2) INTO R3, C(R1) INTO R2
33	PROCEED WITHOUT DATA
34	TERMINATE CURRENT TEST OR LOAD REQUEST
35	TEST LIGHTS
36	FRESH START
37	CHANGE MAJOR MODE

EXTENDED VERBS
(Command Module)

40	ZERO (USED WITH NOUN 20 ONLY)
41	COARSE ALIGN (USED WITH NOUN 20 OR 55 ONLY)
42	FINE ALIGN IMU
43	LOAD IMU ATTITUDE ERROR METERS
44	ILLEGAL VERB
45	ILLEGAL VERB
46	ILLEGAL VERB
47	PERFORM CMS AND SATURN INTEGRATED TEST
50	PLEASE PERFORM
51	PLEASE MARK
52	PERFORM PRELAUNCH ALIGNMENT OPTICAL VERIFICATION
53	ILLEGAL VERB
54	PULSE TORQUE GYROS
55	ALIGN TIME
56	PERFORM BANKSUM
57	PERFORM SYSTEM TEST
60	PREPARE FOR CGC STANDBY
61	RECOVER FROM CGC STANDBY
62	SCAN CSM INBITS
63	ILLEGAL VERB
64	ILLEGAL VERB
65	ILLEGAL VERB
66	ILLEGAL VERB
67	ILLEGAL VERB
70	ILLEGAL VERB
71	ILLEGAL VERB
72	ILLEGAL VERB
73	ILLEGAL VERB
74	ILLEGAL VERB
75	ILLEGAL VERB
76	ILLEGAL VERB
77	ILLEGAL VERB

NORMAL NOUNS
(Command Module)

		SCALE AND DECIMAL POINT
00	NOT IN USE	
01	SPECIFY MACHINE ADDRESS (FRACTIONAL)	(. XXXXX)
02	SPECIFY MACHINE ADDRESS (WHOLE)	(XXXXX.)
03	SPECIFY MACHINE ADDRESS (DEGREES)	(XXX. XXDEGREES)
04	SPECIFY MACHINE ADDRESS (HOURS)	(XXX. XXHOURS)
05	SPECIFY MACHINE ADDRESS (SECONDS)	(XXX. XXSECONDS)
06	SPECIFY MACHINE ADDRESS (GYRO DEGREES)	(XX. XXXDEGREES)
07	SPECIFY MACHINE ADDRESS (T OPT DEGREES)	(XX. XXXDEGREES)
10	CHANNEL TO BE SPECIFIED	
11	SPARE	
12	SPARE	
13	SPARE	
14	SPARE	
15	INCREMENT MACHINE ADDRESS	(OCTAL ONLY)
16	TIME SECONDS	(XXX. XXSECONDS)
17	TIME HOURS	(XXX. XXHOURS)
20	ICDU	(XXX. XXDEGREES)
21	PIPAS	(XXXXX. PULSES)
22	NEW ANGLES I	(XXX. XXDEGREES)
23	DELTA ANGLES I	(XXX. XXDEGREES)
24	DELTA TIME (SECONDS)	(XXX. XXSECONDS)
25	CHECKLIST	(XXXXX.)
26	(PRIO/DELAY, ADRES, BBCON	(OCTAL ONLY)
27	SELF TEST ON/OFF SWITCH	(XXXXX.)
30	STAR NUMBERS	(XXXXX.)
31	FAILREG, SFAIL, ERCOUNT	(OCTAL ONLY)
32	DECISION TIME (MIDCOURSE)	XXX. XXHOURS (INTERNAL UNITS = WEEKS)
33	EPHEMERIS TIME (MIDCOURSE)	XXX. XXHOURS (INTERNAL UNITS = WEEKS)
34	MEASURED QUANTITY (MIDCOURSE)	(XXXX. XKILOMETERS)
35	INBIT MESSAGE	(OCTAL ONLY)
36	LANDMARK DATA 1	(OCTAL ONLY)
37	LANDMARK DATA 2	(OCTAL ONLY)
40	SPARE	
41	SPARE	
42	SPARE	
43	SPARE	
44	SPARE	
45	SPARE	
46	SPARE	
47	SPARE	
50	SPARE	
51	SPARE	(. BBXXXXXMILLIRAD/SEC)
53	GYRO INPUT AXIS ACCELERATION DRIFT	(. BBXXXXX(MILLIRAD/SEC)/(CM/SEC SEC))
54	GYRO SPIN AXIS ACCELERATION DRIFT	(. BBXXXXX(MILLIRAD/SEC)/(CM/SEC SEC))

MIXED NOUNS
(Command Module)

		SCALE AND DECIMAL POINT
55	OCDU	(XXX. XXDEG, XX. XXXDEG)
56	UNCALLED MARK DATA (OCDU & TIME (SECONDS))	(XXX. XXDEG, XX. XXXDEG, XXX. XXSEC)
57	NEW ANGLES OCDU	(XXX. XXDEG, XX. XXXDEG)
60	DELTA GYRO ANGLES FOR PRELAUNCH VERIFICATION	(XXX. XXXDEGREES)
61	TARGET AZIMUTH AND LEVATION	(XXX. XXDEG, XX. XXXDEG)
62	ICDUZ AND TIME	(XXX. XXDEG, XXX. XXSEC)
63	OCDUS AND TIME	(XXX. XXDEG, XXX. XXSEC)
64	OCDUT AND TIME	(XX. XXXDEG, XXX. XXSEC)
65	SAMPLED TIME (HOURS AND SECONDS) (FETCHED IN INTERRUPT)	(XXX. XXHOURS, XXX. XXSEC)
66	SYSTEM TEST RESULTS	(XXXXX. , . XXXXX. XXXXX.)
67	DELTA GYRO ANGLES	(XX. XXXDEG FOR EACH)
70	PIPA BIAS	(X. XXXXCM/SEC SEC FOR EACH)
71	PIPA SCALE FACTOR ERROR	(XXXXX. PARTS/MILLION FOR EACH)
72	DELTA POSITION	(XXXX. XKILOMETERS FOR EACH)
73	DELTA VELOCITY	(XXXX. XMETERS/SEC FOR EACH)
74	MEASUREMENT DATA (MIDCOURSE)	(XXX. XXHOURS (INTERNAL UNITS = WEEKS), XXXX. XKILOMETERS, XXXXX.)
75	MEASUREMENT DEVIATIONS (MIDCOURSE)	(XXXX. XKILOMETERS, XXXX. XMETERS/SEC, XXXX. XKILOMETERS)
76	POSITION VECTOR	(XXXX. XKILOMETERS FOR EACH)
77	VELOCITY VECTOR	(XXXX. XMETERS/SEC FOR EACH)

REGULAR VERBS
(Lunar Module)

01	DISPLAY OCTAL COMP 1 (R1)
02	DISPLAY OCTAL COMP2 (R1)
03	DISPLAY OCTAL COMP 3 (R1)
04	DISPLAY OCTAL COMP 1, 2 (R1, R2)
05	DISPLAY OCTAL COMP 1, 2, 3 (R1, R2, R3)
06	DECIMAL DISPLAY
07	DP DECIMAL DISPLAY (R1, R2)
10	SPARE
11	MONITOR OCT COMP 1 (R1)
12	MONITOR OCT COMP 2 (R1)
13	MONITOR OCT COMP 3 (R1)
14	MONITOR OCT COMP 1, 2, (R1)
15	MONITOR OCT COMP 1, 2, 3, (R1, R2, R3)
16	MONITOR DECIMAL
17	MONITOR DP DECIMAL (R1, R2)
20	SPARE
21	LOAD COMP 1 (R1)
22	LOAD COMP 2 (R2)
23	LOAD COMP 3 (R3)
24	LOAD COMP 1, 2 (R1, R2)
25	LOAD COMP 1, 2, 3 (R1, R2, R3)
26	SPARE
27	FIXED MEMORY DISPLAY
30	REQUEST EXECUTIVE
31	REQUEST WAITLIST
32	C(R2) INTO R3, C(R1) INTO R2
33	PROCEED WITHOUT DATA
34	TERMINATE CURRENT TEST OR LOAD REQUEST
35	TEST LIGHTS
36	FRESH START
37	CHANGE MAJOR MODE

EXTENDED VERBS
(Lunar Module)

40	ZERO (USED WITH NOUNS 20, 40, AND 70)
41	COARSE ALIGN (USED WITH NOUNS 20, 40, AND 70)
42	FINE ALIGN IMU
43	LOAD IMU ATTITUDE ERROR METERS
44	ILLEGAL VERB
45	COMMAND LR TO POSITION 2
46	SAMPLE RADAR ONCE PER SECOND
47	PERFORM LEM FCS TEST
50	PLEASE PERFORM
51	PLEASE MARK
52	PLEASE MARK Y
53	PLEASE MARK X OR Y
54	PULSE TORQUE GYROS
55	ALIGN TIME
56	PERFORM BANK SUM
57	PERFORM SYSTEM TEST
60	ILLEGAL VERB
61	ILLEGAL VERB
62	SCAN LEM INBITS
63	INITIALIZE AGS
64	ILLEGAL VERB
65	ILLEGAL VERB
66	FRESH START FOR LEM SYSTEMS WITH OPTICAL TRACKER
67	ILLEGAL VERB
70	ACQUIRE WITH OPTICAL TRACKER
71	OPTICAL TRACKER SELF TEST
72	RETURN OPTICAL TRACKER TO STOW
73	ILLEGAL VERB
74	ILLEGAL VERB
75	ILLEGAL VERB
76	ILLEGAL VERB
77	ILLEGAL VERB

NORMAL NOUNS
(Lunar Module)

		SCALE AND DECIMAL POINT
00	NOT IN USE	
01	SPECIFY MACHINE ADDRESS (FRACTIONAL)	(.XXXXX)
02	SPECIFY MACHINE ADDRESS (WHOLE)	(XXXXX.)
03	SPECIFY MACHINE ADDRESS (DEGREES)	(XXX. XXDEGREES)
04	SPECIFY MACHINE ADDRESS (HOURS)	(XXX. XXHOURS)
05	SPECIFY MACHINE ADDRESS (SECONDS)	(XXX. XXSECONDS)
06	SPECIFY MACHINE ADDRESS (GYRO DEGREES)	(XX. XXXDEGREES)
07	SPARE	
10	CHANNEL TO BE SPECIFIED	
11	SPARE	
12	SPARE	
13	SPARE	
14	SPARE	
15	INCREMENT MACHINE ADDRESS	(OCTAL ONLY)
16	TIME SECONDS	(XXX. XXSECONDS)
17	TIME HOURS	(XXX. XXHOURS)
20	ICDU	(XXX. XXDEGREES)
21	PIPAS	(XXXXX. PULSES)
22	NEW ANGLES I	(XXX. XXDEGREES)
23	DELTA ANGLES I	(XXX. XXDEGREES)
24	DELTA TIME (SECONDS)	(XXX. XXSECONDS)
25	CHECKLIST	(XXXXX.)
26	PRIO/DELAY, ADRES, BBCON	(OCTAL ONLY)
27	SELF TEST ON/OFF SWITCH	(XXXXX.)
30	STAR NUMBERS	(XXXXX.)
31	FAILREG, SFAIL, ERCOUNT	(OCTAL ONLY)
32	DECISION TIME (MIDCOURSE)	(XXX. XXHOURS (INTERNAL UNITS = WEEKS))
33	EPHEMERIS TIME (MIDCOURSE)	(XXX. XXHOURS (INTERNAL UNITS = WEEKS))
34	MEASURED QUANTITY (MIDCOURSE)	(XXXX. XKILOMETERS)
35	INBIT MESSAGE	(OCTAL ONLY)
36	LANDMARK DATA 1	(OCTAL ONLY)
37	LANDMARK DATA 2	(OCTAL ONLY)
40	RENDEZVOUS RADAR ANGLES (TRUNION, SHAFT)	(XXX. XXDEGREES)
41	NEW RENDEZVOUS RADAR ANGLES (TRUNION, SHAFT)	(XXX. XXDEGREES)
42	AOT ROTATION ANGLES	(XXX. XXDEGREES)
43	AOT DETENT CODE	(XXXXX.)
44	FORWARD VELOCITY, LATERAL VELOCITY	(XXXXX. FEET/SEC)
45	ROTATIONAL HAND CONTROLLER ANGLE RATES	(XXXXX. DEG/SEC)
46	SPARE	
47	SPARE	
50	SPARE	
51	SPARE	
52	GYRO BIAS DRIFT	
53	GYRO INPUT AXIS ACCELERATION DRIFT	(.BBXXXXXMILLIRAD/SEC)/(CM/SEC SEC))
54	GYRO SPIN AXIS ACCELERATION DRIFT	(.BBXXXXX(MILLIRAD/SEC)/(CM/SEC SEC))

MIXED NOUNS
(Lunar Module)

		SCALE AND DECIMAL POINT
55	LANDING RADAR ALTITUDE, TIME (SECONDS)	(XXXXX. FEET, XXX. XXSEC)
56	LANDING RADAR VELX, TIME (SECONDS)	(XXXXX. FEET/SEC, XX. XXSEC)
57	LANDING RADAR VELY, TIME (SECONDS)	(XXXXX. FEET/SEC, XXX. XXSEC)
60	LANDING RADAR VELZ, TIME (SECONDS)	(XXXXX. FEET/SEC, XXX. XXSEC)
61	TARGET AZIMUTH AND ELEVATION	(XXX. XXDEG, XX. XXDEG)
62	RENDEZVOUS RADAR RANGE, TRUNION, SHAFT	(XXXXXB. FEET, XXX. XXDEG, XX. XXDEG)
63	RENDEZVOUS RADAR RANGE RATE, TRUNION, SHAFT	(XXXXX. FEET/SEC, XX. XXDEG, XX. XXDEG)
64	INITIAL ALTITUDE, FINAL ALTITUDE, ALT. RATE	(XXXXX. FEET, XXXXX. FEET, XXXXX. FEET/SEC)
65	SAMPLED TIME (HOURS AND SECONDS) (FETCHED IN INTERRUPT)	(XXX. XXHOURS, XXX. XXSEC)
66	SYSTEM TEST RESULTS	(XXXXX, , XXXXX, XXXXX,)
67	DELTA GYRO ANGLES	(XX. XXDEG FOR EACH)
70	OPTICAL TRACKER ANGLES (AZIMUTH, ELEVATION)	(XXX. XXDEG, XXX. XXDEG)
71	DESIRED OPTICAL TRACKER ANGLES (AZ, ELEV)	(XXX. XXDEG, XXX. XXDEG)
72	DELTA POSITION	(XXXX. XKILOMETERS FOR EACH)
73	DELTA VELOCITY	(XXXX. XMETERS/SEC FOR EACH)
74	MEASUREMENT DATA (MIDCOURSE)	(XXX. XXHOURS (INTERNAL UNITS = WEEKS), XXXX. XKILOMETERS, XXXXX.
75	MEASUREMENT DEVIATIONS (MIDCOURSE)	(XXXX. XKILOMETERS, XXXX. XMETERS/SEC, XXXX. XKILOMETERS)
76	POSITION VECTOR	(XXXX. XKILOMETERS FOR EACH)
77	VELOCITY VECTOR	(XXXX. XMETERS/SEC FOR EACH)

MEMORY ADDRESSING

ARRANGEMENT OF ADDRESSES

OCTAL PSEUDO-ADDRESS	REGISTER NAME	REMARKS	
0000	A		
0001	L	(also channel 01)	
0002	Q	(also channel 02)	
0003	EB	Erasable Bank Register	Flip-Flop
0004	FB	Fixed Bank Register	registers
0005	Z		
0006	BB	Both Bank Registers	
0007	--	Zeros	
0010	ARUPT	xRUPT = Storage for x	
0011	LRUPT	during Interrupt;	
0012	QRUPT	ZRUPT & BRUPT	
0013	(spare)	stored automatically.	
0014	(spare)		
0015	ZRUPT		
0016	BBRUPT		
0017	BRUPT		
			2040 words
0020	CYR	Cycle Right 1 Bit	of Erasable
0021	SR	Shift Right 1 Bit	
0022	CYL	Cycle Left 1 Bit	
0023	EDOP	Edit Opcode	
0024-0060	Counters		
0061-1377	Unswitched Erasable		
1400-3777	5 Erasable Banks at 256 words		
4000-up	Fixed		Fixed

FIXED & ERASABLE BANK - SWITCHING

OCTAL PSEUDO-ADDRESS	MEMORY TYPE	ERASABLE BANK REG.	FIXED BANK REG.	FIXED EXTENSION BIT (CHANNEL 7)	S-REG VALUE
00000-01377	(Note 1)	x	xx	x	0000-1377
00000-00377	(Note 1)	0	xx	x	1400-1777
00400-00777	Unswitched E	1	xx	x	1400-1777
01000-01377	Unswitched E	2	xx	x	1400-1777
01400-01777	Switched E	3	xx	x	1400-1777
02000-02377	Switched E	4	xx	x	1400-1777
02400-02777	Switched E	5	xx	x	1400-1777
03000-03377	Switched E	6	xx	x	1400-1777
03400-03777	Switched E	7	xx	x	1400-1777
04000-07777	Fixed-fixed	x	xx	x	4000-7777
10000-11777	Common fixed	x	00	x	2000-3777
12000-13777	Common fixed	x	01	x	2000-3777
14000-15777	Fixed-fixed	x	02	x	2000-3777
16000-17777	Fixed-fixed	x	03	x	2000-3777
20000-21777	Common fixed	x	04	x	2000-3777
22000-23777	Common fixed	x	05	x	2000-3777
		-- and so on through:			
64000-65777	Common fixed	x	26	x	2000-3777
66000-67777	Common fixed	x	27	x	2000-3777
70000-71777	Super-bank 0	x	30	0	2000-3777
72000-73777	Super-bank 0	x	31	0	2000-3777
		-- and so on through:			
106000-107777	Super-bank 0	x	37	0	2000-3777
110000-11177	Super-bank 1	x	30	1	2000-3777
112000-113777	Super-bank 1	x	31	1	2000-3777
114000-115777	Super-bank 1	x	32	1	2000-3777
116000-117777	Super-bank 1	x	33	1	2000-3777

INPUT AND OUTPUT CHANNELS

OUTPUT CHANNELS 05 AND 06

RCS Control Signals

Bit Position	CMC		LGC
Channel 05	SM Mode	CM Mode	
1	RC+X+P	+PCH/-X/+YAW	-X/-P/+R
2	RC-X-P	-PCH/+Z	+X/+P/-R
3	RC-X+P	+PCH/-X/-YAW	-X/+R/+P
4	RC+X-P	-PCH/-Z	+X/+R/-P
5	RC+X+Y	+YAW/-X/+PCH	-X/+R/-R
6	RC-X-Y	-YAW/-X/-PCH	+X/-P/+R
7	RC-X+Y	+YAW/-X/-PCH	-X/-R/-P
8	RC+X-Y	-YAW/-X/+PCH	+X/+R/+P
Channel 06			
1	RC+Z+R	+RLL/(+Y, +Z)	+Z/+YAW
2	RC-Z-R	-RLL/(-Y, -Z)	-Z/-YAW
3	RC-Z+R	+RLL/(+Y, -Z)	-Z/+YAW
4	RC+Z-R	-RLL/(-Y, +Z)	+Z/-YAW
5	RC+Y+R	Not used	+Y/+YAW
6	RC-Y-R	Not used	-Y/-YAW
7	RC-Y+R	Not used	-Y/+YAW
8	RC+Y-R	Not used	+Y/-YAW

OUTPUT CHANNEL 10

Bit	CMC	LGC
1	RLYB 1	Same as CMC
2	RLYB 2	
3	RLYB 3	
4	RLYB 4	
5	RLYB 5	
6	RLYB 6	
7	RLYB 7	
8	RLYB 8	
9	RLYB 9	
10	RLYB 10	
11	RLYB 11	
12	Relay Address 1	
13	Relay Address 2	
14	Relay Address 3	
15	Relay Address 4	

OUTPUT CHANNEL 11

Bit	CMC	LGC
1	ISS Warning	Same as CMC
2	Light Comp Activity Lamp	
3	Light Uplink Activity Lamp	
4	Light Temp Caution Lamp	
5	Light Keyboard Release Lamp	
6	Flash Verb and Noun Lamps	
7	Light Operator Error Lamp	
8	Spare	
9	Test Connector Outbit	
10	Caution Reset	
11	Spare	
12	Spare	
13	Engine On	
14	Engine Off	
15	Spare	

OUTPUT CHANNEL 14

Bit Position	CMC	LGC
1	Transmit An Outlink Bit	Same As In CMC
2	Spare	Altitude Rate Selection
3	Spare	Transmit An Altitude Meter Bit
4	Spare	Drive Thrust
5	Drive EMS	Spare
6	Enable Gyro	Same As CMC
7	Gyro Selection b	
8	Gyro Selection a	
9	Gyro Selection c (minus sign)	
10	Drive Gyro	

Gyro Selection c a b	Signals Generated By Gyro Drive Control
0 0 0	none
1 0 0	none
0 0 1	GYXP Drive X Gyro Positive
1 0 1	GYXM Drive X Gyro Negative
0 1 0	GYYP Drive Y Gyro Positive
1 1 0	GYYM Drive Y Gyro Negative
0 1 1	GYZP Drive Z Gyro Positive
1 1 1	GYZM Drive Z Gyro Negative

11	Drive Optics CDU S	Same As CMC
12	Drive Optics CDU T	
13	Drive IMU CDU Z	
14	Drive IMU CDU Y	
15	Drive IMU CDU X	

INPUT CHANNEL 15

Bit Position	CMC	LGC
1	KEY 1	KEY 1
2	KEY 2	KEY 2
3	KEY 3	KEY 3
4	KEY 4	KEY 4
5	KEY 5 — From Main Panel DSKY	KEY 5 — From LEM DSKY

INPUT CHANNEL 16

Bit Position	CMC	LGC
1	KEY 1	-
2	KEY 2	-
3	KEY 3 — From Navigation Panel DSKY	MARK X
4	KEY 4	MARK Y
5	KEY 5	MARK REJECT
6	MARK — From Optics	DESCENT +
7	MARK REJECT	DESCENT -

OUTPUT CHANNEL 12

Bit Position	CMC	LGC
1	Zero Optics CDU	Zero Rendezvous Radar CDU
2	Enable Optics Error Counter	Enable Rendezvous Radar Counter
3	Star Tracker On	Use Low Scale Factor For Horizon Velocity
4	Enable Coarse Alignment	Same As CMC
5	Zero IMU CDU's	Same As CMC
6	Enable IMU Error Counter	Same As CMC
7	Spare	Spare
8	SPS TVC Enable	Display Inertial Data
9	Enable SIVB Take Over	Trim Plus Pitch Gimbal
10	Zero Optics	Trim Minus Pitch Gimbal
11	Disengage Optics DAC	Trim Plus Roll Pitch Gimbal
12	Spare	Trim Minus Roll Pitch Gimbal
13	Enable SIV Start Sequence	Accept Landing Radar Position Command
14	Cutoff SIV Engine	Enable Rendezvous Radar Lock On
15	ISS Turn On Delay Completed	Same As CMC

OUTPUT CHANNEL 13

Bit Position	CMC	LGC
1	Spare	Radar Mode Selection c
2	Spare	Radar Mode Selection b
3	Spare	Radar Mode Selection a
4	Spare	Transmit a Control Signal To Radar

a	b	c	Signals Generated By Radar Control (Used In LM Only)
0	0	0	None
0	0	1	RRRANG Supply Rendezvous Radar Range Data
0	1	0	RRRARA Supply Rendezvous Radar Range Rate
0	1	1	None
1	0	0	LRXVEL Supply Landing Radar X Velocity Data
1	0	1	LRYVEL Supply Landing Radar Y Velocity Data
1	1	0	LRZVEL Supply Landing Radar Z Velocity Data
1	1	1	LRRANG Supply Landing Radar Range Data

5	Inhibit Uplink Enable Crosslink	Same As CMC
6	Block Inlink	
7	Downlink Order Bit	
8	Enable BMAG Counter	Enable RHC Counter
9	Spare	Start RHC Read
10	Test Alarm	Same As CMC
11	Enable Standby	
12	Reset Trap 31-A	
13	Reset Trap 31-B	
14	Reset Trap 32	
15	Enable T6RUPT	

INPUT CHANNEL 30

Bit Position	CMC	LGC
1	Ullage Thrust Present	Abort
2	Service Module Separation	Stage Verification
3	SPS Ready	Engine Armed
4	SIVB Separation	Stage Abort
5	Lift Off SIVB	Auto Throttle
6	Guidance Reference Release	Inertial Data Display
7	Optics CDU Fail	Rendezvous Radar CDU Fail
8	Spare	Spare
9	IMU Operate	Same As CMC
10	Saturn Control	G&N Control
11	IMU Caged	Same As CMC
12	IMU CDU Fail	
13	IMU Fail	
14	ISS Turn On Request	
15	Temperature Within Limits	

INPUT CHANNEL 31

Bit Position	CMC	LGC
1	+ Pitch Command From Manual Rotation Control	+EL (LPD) + PM1
2	- Pitch Command From Manual Rotation Control	-EL (LPD) - PM1
3	+ Yaw Command From Manual Rotation Control	+ YM1
4	- Yaw Command From Manual Rotation Control	- YM1
5	+ Roll Command From Manual Rotation Control	+ AZ (LPD) + RM1
6	- Roll Command From Manual Rotation Control	- AZ (LPD) - RM1
7	+ X Translation Command From Manual Translation Control	Same As CMC
8	- X Translation Command From Manual Translation Control	
9	+ Y Translation Command From Manual Translation Control	
10	- Y Translation Command From Manual Translation Control	
11	+ Z Translation Command From Manual Translation Control	
12	- Z Translation Command From Manual Translation Control	
13	Hold Function	Attitude Hold
14	Free Function	Auto Stabilization
15	Guidance Control, Auto Pilot Control	Attitude Control Out Of Detent

INPUT CHANNEL 32

Bit Position	CMC	LGC
1	+ Pitch Minimum Impulse	Thruster 2-4 Fail
2	- Pitch Minimum Impulse	Thruster 5-8 Fail
3	+ Yaw Minimum Impulse	Thruster 1-3 Fail
4	- Yaw Minimum Impulse	Thruster 6-7 Fail
5	+ Roll Minimum Impulse	Thruster 14-16 Fail
6	- Roll Minimum Impulse	Thruster 13-15 Fail
7	Spare	Thruster 9-12 Fail
8	Spare	Thruster 10-11 Fail
9	Spare	Pitch Gimbal Fail
10	Spare	Roll Gimbal Fail
11	LM Attached	Altitude Hold
12	Spare	Spare
13	Spare	Spare
14	Spare	Spare
15	Spare	Spare

INPUT CHANNEL 33

Bit Position	CMC	LGC
1	Spare	Spare
2	Spare	Rendezvous Radar On Automatic
3	Spare	Rendezvous Radar On Low Scale
4	Zero Optics	Rendezvous Radar Data Good
5	AGC Has Control	Landing Radar Data Good
6	Spare	Landing Radar In Position (Descent)
7	Spare	Landing Radar In Position 2 (Hover)
8	Spare	Landing Velocity Data Good
9	Spare	Landing Radar On Low Scale
10	Accept Uplink Data	Same As CMC
11	Inlink Too Fast	
12	Downlink Too Fast	
13	PIPA Fail	
14	AGC Warning	
15	Oscillator Alarm	

CONTROL PULSES

Pulse	Purpose
A2X	Copies bits 16 through 1 of register A directly (not through WA's) into bit positions 16 through 1 of register X.
B15X	Enters a ONE into bit position 15 of register X.
CI	Inserts carry bit into bit position 1 of the Adder. This adds the quantity one to the content of the Adder if no bit is carried around (from bit positions 16 to bit position 1).
CLXC	Clears register X if flip-flop BR1 contains a ZERO. (Used in instruction DV E.)
DVST	Modifies the content of the stage counter (ST) by complementing the content of the next higher bit position as shown below: Binary → Octal 000 → 0 001 → 1 011 → 3 111 → 7 110 → 6 100 → 4
EXT	Enters a ONE into bit position EXT of register SQ.
G2LS	Copies bits 16, 15 through 4, and 1 of register G directly (not through WA's) into bit positions 16, 12 through 1, and 15 of register X.
KRPT	Resets interrupt priority cell.
L16	Enters a ONE into bit position 16 of register L.
L2GD	Copies bits 16 and 14 through 1 of register L directly (not through WA's) into bit positions 16 and 15 through 2 of register G; enters a ONE into bit position 1 of register G if pulse MCRO is generated.
MONEX	Clears register X and enters ONE's into bit positions 16 through 2.
MOUT	Causes the generation of one minus drive pulse.
NEACOF	Permits end around carry upon completion of subinstruction MP3.
NEACON	Inhibits end around carry (also during WYD) until NEACOF.
NISQ	Causes loading of next instruction into register SQ (implies RB and WSQ at time 12). Also resets the stage counter (ST) to 0; frees certain restrictions; permits execution of instruction RUPT and of all Counter Instructions.
PIFL	Prevents writing into bit position 1 of register Y on control pulse WYD if bit position 15 of register L contains a ONE. (Used in instruction DV.)
PONEX	Clears register X and enters a ONE into bit position 1.
POUT	Causes the generation of one plus drive pulse.
PTWOX	Clears register X and enters a ONE into bit position 2.
R15	Enters 000015 into WA's.
R1C	Enters 177776 (minus one) into WA's.
R6	Enters 000006 into WA's.
RA	Reads bits 16 through 1 of register A into WA's 16 through 1.
RAD	Reads address of next instruction. RAD appears at last time S of an instruction and is normally interpreted as RG. If the next instruction is INHINT, RELINT, or EXTEND, RAD is interpreted as RZ and ST2 instead.
RB	Reads bits 16 through 1 of register B into WA's 16 through 1.
RB1	Enters 000001 into WA's.
RB1F	Enters 000001 into WA's if flip-flop BR1 contains a ONE.
RB2	Enters 000002 into WA's.
RBBK	Reads the BB (both bank) configuration into the WA's, i.e., copies the content of bit position 16 of register FBANK into WA's 16 and 15, the content of bit positions 14 through 11 of register FBANK into WA's 14 through 11, and the content of bit positions 11 through 9 of register EBANK into WA's 3 through 1.

Pulse	Purpose
RC	Reads the complemented content of register B (bits 16 through 1 of C) into WA's 16 through 1.
RCH	Reads the content of the input-output channel specified by the contents of register S; bit 15 is read into WA's 16 and 15, and bits 14 through 1 are read into WA's 14 through 1.
RG	Reads bits 16 through 1 of register G and WA's 16 through 1.
RL	Reads bit 16 of register L into WA's 16 and 15, and bits 14 through 1 into WA's 14 through 1.
RL10BB	Reads low 10 bits, i.e., bits 10 through 1 of register B into WA's 10 through 1; replaces c(S), which include a quarter code, by a 10 bit address.
RQ	Reads bits 16 through 1 of register Q into WA's 16 through 1.
RRPA	Enters into the WA's the address of a RUPT Transfer Routine supplied by the Interrupt Priority Control.
RSC	Reads the content of the CP register specified by the content of register S; bits 16 through 1 are read into WA's 16 through 1.
RSCT	Enters into the WA's the address of a counter address supplied by the Counter Priority Control (paragraph 30-94).
RSTRT	Enters 004000 (Block II start address) into WA's.
RSTSTG	Resets the stage counter to 0 (refer to DVST).
RU	Reads bits 16 through 1 of Adder output gates (U) into WA's 16 through 1.
RUS	Reads bit 15 of Adder output gates (U) into WA's 16 and 15, and bits 14 through 1 into WA's 14 through 1.
RZ	Reads bits 16 through 1 of register Z into WA's 16 through 1.
ST1	Sets stage 1 flip-flop to ONE at next time 12.
ST2	Sets stage 2 flip-flop to ONE at next time 12.
STAGE	Causes the execution of next subinstruction as defined by the content of the stage counter (ST).
TL15	Copies bit 15 of register L into flip-flop BR1.
TMZ	Tests the content of the WA's for minus zero: if bits 16 through 1 are all ONE's, flip-flop BR2 is set to ONE; otherwise BR2 is set to ZERO.
TOV	Tests the content of WA's 16 and 15 for overflow: set flip-flops BR1 and BR2 to 01 in case of positive overflow, or to 10 in case of negative overflow.
TPZG	Tests the content of register G for plus zero: if bits 16 through 1 are all ZERO's, flip-flop BR2 is set to ONE; otherwise the content of BR2 is not changed.
TRSM	Tests signals XT1/ and XB7 of selection logic for the resume address (0017) during the execution of subinstruction NDX0; if 0017 is present, subinstruction RSM3 is executed next by setting c(ST) = 3; otherwise subinstruction NDX1 by setting c(ST) = 1.
TSGN	Tests content of WA 16 for sign: if a ZERO, flip-flop BR1 is set to ZERO; if a ONE, flip-flop BR1 is set to ONE without changing the content of flip-flop BR2.
TSGN2	Tests content of WA 16 for sign: if a ZERO, flip-flop BR2 is set to ZERO; if a ONE, flip-flop BR2 is set to ONE without changing the content of flip-flop BR1.
TSGU	Tests content of output gate U16 of Adder for sign: if a ZERO, flip-flop BR1 is set to ZERO; if a ONE, flip-flop BR1 is set to ONE.
U2BBK	Copies bits 16 and 14 through 11 of the Adder output gates (U) into bit positions 16 and 14 through 11 of register FBANK, and bits 3 through 1 (of U) into bit positions 3 through 1 of register EBANK. U2BBK may be inhibited by signal MONWBK, which is generated if register BBANK is addressed.
WA	Clears register A and writes the content of WA's 16 through 1 into bit positions 16 through 1.

Pulse	Purpose
WALS	Clears register A and writes the content of WA's 16 through 3 into bit positions 14 through 1. If bit position 1 of register G contains a ZERO, the content of bit position 16 of register G is entered into bit positions 16 and 15 of register A; if bit position 1 of register G contains a ONE, the content of output gate U 16 of the Adder is entered into bit positions 16 and 15 of register A. WALS also clears bit positions 14 and 13 of register L, and writes the content of WA's 2 and 1 into these bit positions.
WB	Clears register B and writes the content of WA's 16 through 1 into bit positions 16 through 1.
WCH	Clears the output channel specified by the content of register S and writes the content of WA's 16 and 14 through 1 into bit positions 15 through 1.
WG	Clears register G and writes the content of WA's 16 through 1 into bit positions 16 through 1, except if register S contains address 0020 through 0023 in which case the WA content is cycled or shifted (paragraph 30-41).
WL	Clears register L and writes the content of WA's 16 through 1 into bit positions 16 through 1.
WOVR	Tests the content of WA's 16 and 15 for positive overflow: if register S contains 0025, counter 0024 is incremented; if register S contains 0026, 0027, or 0030, instruction RUPT is executed.
WQ	Clears register Q and writes the content of WA's 16 through 1 into bit positions 16 through 1.
WS	Clears register S and writes the content of WA's 12 through 1 into bit positions 12 through 1.
WSC	Clear the CP register specified by the content of register S and writes the content of WA's 16 through 1 into bit positions 16 through 1.
WSQ	Clears register SQ and writes the content of WA's 16 and 14 through 10 into bit positions 16 and 14 through 10.
WY	Clears registers X and Y and carry flip-flop CI; writes the content of WA's 16 through 1 into bit positions 16 through 1 of register Y.
WY12	Clears registers X and Y and carry flip-flop CI; writes the content of WA's 12 through 1 into bit positions 12 through 1 of register Y.
WYD	Clears registers X and Y and carry flip-flop CI, writes the content of WA's 16 and 14 through 1 into bit positions 16 and 15 through 2 of register Y; writes the content of WA 16 into bit position 1 of register Y except in SHINC sequence, or unless bit position 15 of register L contains a ONE at PIFL, or if end around carry is inhibited by control pulse NEACON.
WZ	Clears register Z and writes the content of WA's 16 through 1 into bit positions 16 through 1.
Z15	Enters a ONE into bit position 15 of register Z.
Z16	Enters a ONE into bit position 16 of register Z.
ZAP	Causes the generation of control pulses RU, G2LS, and WALS (used in instruction MP K).
ZIP	Causes generation of control pulses A2X and L2GD (used in instruction MP K): performs read/write operations depending on the content of bit positions 15, 2, and 1 of register L as shown:

L15	L2	L1	Read	Write	Carry	Remember
0	0	0	-	WY	-	-
0	0	1	RB	WY	-	-
0	1	0	RB	WYD	-	-
0	1	1	RC	WY	CI	MCRO
1	0	0	RB	WY	-	-
1	0	1	RB	WYD	-	-
1	1	0	RC	WY	CI	MCRO
1	1	1	-	WY	-	MCRO

If MCRO occurs, a ONE is entered into L15

ZOUT	Stops the generation of drive pulses.

Silver, G.L.