

Next +160 bits, 2-10 and shift + them + two right in AC

408

VALU

112

14-JUN-74

1614

PAGE 1-8

(10)

Call subroutine. Port to proto; WORD2 according to user's desire.

WORD1 = complement [WORD2]. This is so we can print names.

DF micro-instruction. Folk names could be "SOPA CLA"

1274 1444 AMD NS710  
1274 7112 CLL RTH  
1275 1460 TAU HINUM  
1275 3464 JDS T-PROT

TAD WORD2

CMA DCA WORD2

TAU HINUM

For Doloop SP1 22 = -25 = - [number of micro-instruction-instruction]

1275 4113 UCA X

AU RET25

1276 3116 TCA ZL  
1277 1513 TAU I X  
1340 9873 AND WORD1  
1391 1461 CIA TAU WORD1  
1392 1623 TAU WORD1  
1393 2113 ISF L

S2A CLA

NO WHICH

NO I

NO WORD1

ISF L

NAME ASSOCIATED WITH [I X]

At the new location [I X] is a 1 in every bit where it is permissible for WORD1 to have both 1-bit and also have the name associated with [I X]. We check if WORD1 meets these conditions. If no JMP WHICH and consider another name. If yes continue. Also whether yes or no increment X here. This saves 2 word comparisons.

1394 7644 TAU WORD1

1395 4922 S2A CLA

1396 1460 NO WHICH

1397 1624 TAU WORD1

1398 2113 TAU WORD1

1399 2113 TAU WORD1

1400 2113 TAU WORD1

1401 2113 TAU WORD1

1402 2113 TAU WORD1

1403 2113 TAU WORD1

1404 2113 TAU WORD1

1405 2113 TAU WORD1

1406 2113 TAU WORD1

1407 2113 TAU WORD1

1408 2113 TAU WORD1

1409 2113 TAU WORD1

1410 2113 TAU WORD1

1411 2113 TAU WORD1

1412 2113 TAU WORD1

1413 2113 TAU WORD1

1414 2113 TAU WORD1

1415 2113 TAU WORD1

1416 2113 TAU WORD1

1417 2113 TAU WORD1

1418 2113 TAU WORD1

1419 2113 TAU WORD1

1420 2113 TAU WORD1

1421 2113 TAU WORD1

1422 2113 TAU WORD1

1423 2113 TAU WORD1

1424 2113 TAU WORD1

1425 2113 TAU WORD1

1426 2113 TAU WORD1

1427 2113 TAU WORD1

1428 2113 TAU WORD1

1429 2113 TAU WORD1

1430 2113 TAU WORD1

1431 2113 TAU WORD1

1432 2113 TAU WORD1

1433 2113 TAU WORD1

1434 2113 TAU WORD1

1435 2113 TAU WORD1

1436 2113 TAU WORD1

1437 2113 TAU WORD1

1438 2113 TAU WORD1

1439 2113 TAU WORD1

1440 2113 TAU WORD1

1441 2113 TAU WORD1

1442 2113 TAU WORD1

1443 2113 TAU WORD1

1444 2113 TAU WORD1

1445 2113 TAU WORD1

1446 2113 TAU WORD1

1447 2113 TAU WORD1

1448 2113 TAU WORD1

1449 2113 TAU WORD1

1450 2113 TAU WORD1

1451 2113 TAU WORD1

1452 2113 TAU WORD1

1453 2113 TAU WORD1

1454 2113 TAU WORD1

1455 2113 TAU WORD1

1456 2113 TAU WORD1

1457 2113 TAU WORD1

1458 2113 TAU WORD1

1459 2113 TAU WORD1

1460 2113 TAU WORD1

1461 2113 TAU WORD1

1462 2113 TAU WORD1

1463 2113 TAU WORD1

1464 2113 TAU WORD1

1465 2113 TAU WORD1

1466 2113 TAU WORD1

1467 2113 TAU WORD1

1468 2113 TAU WORD1

1469 2113 TAU WORD1

1470 2113 TAU WORD1

1471 2113 TAU WORD1

1472 2113 TAU WORD1

1473 2113 TAU WORD1

1474 2113 TAU WORD1

1475 2113 TAU WORD1

1476 2113 TAU WORD1

1477 2113 TAU WORD1

1478 2113 TAU WORD1

1479 2113 TAU WORD1

1480 2113 TAU WORD1

1481 2113 TAU WORD1

1482 2113 TAU WORD1

1483 2113 TAU WORD1

1484 2113 TAU WORD1

1485 2113 TAU WORD1

1486 2113 TAU WORD1

1487 2113 TAU WORD1

1488 2113 TAU WORD1

1489 2113 TAU WORD1

1490 2113 TAU WORD1

1491 2113 TAU WORD1

1492 2113 TAU WORD1

1493 2113 TAU WORD1

1494 2113 TAU WORD1

1495 2113 TAU WORD1

1496 2113 TAU WORD1

1497 2113 TAU WORD1

1498 2113 TAU WORD1

1499 2113 TAU WORD1

1500 2113 TAU WORD1

1501 2113 TAU WORD1

1502 2113 TAU WORD1

1503 2113 TAU WORD1

1504 2113 TAU WORD1

1505 2113 TAU WORD1

1506 2113 TAU WORD1

1507 2113 TAU WORD1

1508 2113 TAU WORD1

1509 2113 TAU WORD1

1510 2113 TAU WORD1

1511 2113 TAU WORD1

1512 2113 TAU WORD1

1513 2113 TAU WORD1

1514 2113 TAU WORD1

1515 2113 TAU WORD1

1516 2113 TAU WORD1

1517 2113 TAU WORD1

1518 2113 TAU WORD1

1519 2113 TAU WORD1

1520 2113 TAU WORD1

1521 2113 TAU WORD1

1522 2113 TAU WORD1

1523 2113 TAU WORD1

1524 2113 TAU WORD1

1525 2113 TAU WORD1

1526 2113 TAU WORD1

1527 2113 TAU WORD1

1528 2113 TAU WORD1

1529 2113 TAU WORD1

1530 2113 TAU WORD1

1531 2113 TAU WORD1

1532 2113 TAU WORD1

1533 2113 TAU WORD1

1534 2113 TAU WORD1

1535 2113 TAU WORD1

1536 2113 TAU WORD1

1537 2113 TAU WORD1

1538 2113 TAU WORD1

1539 2113 TAU WORD1

1540 2113 TAU WORD1

1541 2113 TAU WORD1

1542 2113 TAU WORD1

1543 2113 TAU WORD1

1544 2113 TAU WORD1

1545 2113 TAU WORD1

1546 2113 TAU WORD1

1547 2113 TAU WORD1

1548 2113 TAU WORD1

1549 2113 TAU WORD1

1550 2113 TAU WORD1

1551 2113 TAU WORD1

1552 2113 TAU WORD1

1553 2113 TAU WORD1

1554 2113 TAU WORD1

1555 2113 TAU WORD1

1556 2113 TAU WORD1

1557 2113 TAU WORD1

1558 2113 TAU WORD1

1559 2113 TAU WORD1

1560 2113 TAU WORD1

1561 2113 TAU WORD1

1562 2113 TAU WORD1

1563 2113 TAU WORD1

1564 2113 TAU WORD1

1565 2113 TAU WORD1

1566 2113 TAU WORD1

1567 2113 TAU WORD1

1568 2113 TAU WORD1

1569 2113 TAU WORD1

1570 2113 TAU WORD1

1571 2113 TAU WORD1

1572 2113 TAU WORD1

1573 2113 TAU WORD1

1574 2113 TAU WORD1

1575 2113 TAU WORD1

1576 2113 TAU WORD1

1577 2113 TAU WORD1

1578 2113 TAU WORD1

1579 2113 TAU WORD1

1580 2113 TAU WORD1

1581 2113 TAU WORD1

1582 2113 TAU WORD1

1583 2113 TAU WORD1

1584 2113 TAU WORD1

1585 2113 TAU WORD1

1586 2113 TAU WORD1

1587 2113 TAU WORD1

1588 2113 TAU WORD1

1589 2113 TAU WORD1

1590 2113 TAU WORD1

1591 2113 TAU WORD1

1592 2113 TAU WORD1

1593 2113 TAU WORD1

1594 2113 TAU WORD1

1595 2113 TAU WORD1

1596 2113 TAU WORD1

1597 2113 TAU WORD1

1598 2113 TAU WORD1

1599 2113 TAU WORD1

1600 2113 TAU WORD1

1601 2113 TAU WORD1

1602 2113 TAU WORD1

1603 2113 TAU WORD1

1604 2113 TAU WORD1

1605 2113 TAU WORD1

1606 2113 TAU WORD1

1607 2113 TAU WORD1

1608 2113 TAU WORD1

1609 2113 TAU WORD1

1610 2113 TAU WORD1

1611 2113 TAU WORD1

1612 2113 TAU WORD1

1613 2113 TAU WORD1

1614 2113 TAU WORD1

1615 2113 TAU WORD1

1616 2113 TAU WORD1

1617 2113 TAU WORD1

1618 2113 TAU WORD1

1619 2113 T

(21)

418

PAGE

1-8

# CONVERSATION OF XEROX

1263	1642				
1264	7112	ANL 43774			
1265	3720	CAL RTW			
1266	3874	TAD KOPEN			
1267	4324	OCA W002			
1268	1624	JNS I POUT			
1271	7444	TAD VURUP			
1272	3424	CMA			
1273	1613	UCA W001			
1281	7841	TAD MNICOD			
1282	1423	CIA X			
1283	1884	TAD NLC25			
1284	3110	OCA RTW			
1287	1513	MNCL.			
1294	4923	TAD X			
1295	7843	AND W001			
1296	1423	CIA X			
1297	1884	TAD W001			
1298	7644	ISZ X			
1309	5322	SEA DA			
1309	5322	JMP WHICK			
1313	2913	TAD X			
1307	5824	AND W002			
1314	1424	CIA X			
1315	7841	TAD X			
1316	1114	OCA X			
1317	3115	TAD W002			
1324	4579	JHS Y			
1321	5112	NAME			
1322	2113	MNCL.			
1323	5272	ISZ X			
1324	1324	JMS MNCL			
1325	5272	JMS MNCL			
1326	3673	DCB KWDNT	You reach this point if the number has 9 or more digits. The purpose of first box is to determine whether the number is positive or negative. You test to see if [bit #5] = 0 or 1. If 0 then num is positive and make SIGN=0. If =1 then num is negative and Set SIGN = -1		
1327	9342	JMP ITOPC			
1328	7244	W4SHN, STA			
1329	3875	WPSHN, DCASGN			
1330	7241	AND N2778			
1331	6977	CBL RTW			
1332	7241	PRG			
1333	7212	PRG			
1334	7212	PRG			
1335	7212	PRG			
1336	5342	PRG			
1337	7271	MLNUM, CBL HUA			
1338	5342	PRG			
1339	1614	AND N28			
1340	7244	SEA			
1341	7244	STA			
1342	7244	OCA SIGN	You reach this point whether the number has less than 9 digits or more. In either event at the time you reach W002 + the absolute value of number is in AC. Therefore store it in X		
1343	3472	AUT POUT			
1344	1417	UCA X			
1345	3113	MNCL:	The purpose of this box which contains on next page is to		
1346	1113	TAD X	Set WORD2 = X when SIGN=0 i.e. number is positive		
1347	3472	OCA W002	WORD2 = -X when SIGN=-1 i.e. number is negative		
1348	1423	TAD SIGN			
1349	1473	OCA CLA			

↓  
Thus WORD2 = real value of number. X contains + to = absolute

22

PALP  
V1.2

14-JUN-70

↑

SEE

last

page

for

meaning

of

this

box

↓

PAGE

1-70

call

subroutine

POUT

to

process

WORD

according

to user's

needs

↓

TEST

SIGN

to

determine

SKN

of

numb

ber.

If

number

is

positive

use

↓

PRINT

↓

SUBROUTINE

OUT

PRINTS

ONLY

WHEN

SSL=0

I.E.

USER

DESIDES

LISTING

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

↓

2412

23

PAGE IS CONTINUATION OF XEROX PAGE. I USE IT BECAUSE I CAN

	L1	V142	13-JUN-24	16:44
1352	5556		JMP	*4
1353	1113		JAD	X
1354	7441		CIA	
1355	3814		DGA	
1356	6554		WORD?	
1357	1875		JHS	I PUUT
1358			SIGN	
1359	7514		SPA	
1360	5389		JHP	*
1361	42552		JHP	*4
1362	42552		JMS	I DUT
1363	8253		JMS	I DUT
1364	5367		253	"**"
1365	4552		JHP	*3
1366	7245		JHS	OUT
1367	1113		JHS	"**"
1368	4553		TAD	X
1369	4244		JMS	WNUANG
1370	5542		JMS	WUANG
			JHP	TUPAK

/HECAUSE CUD SINAGHT LINE  
/WULSONT MATTER WHERE PAGE CHANGEE  
WTAUL, TAO I PUPHOR

You reach WTADLA two ways. The first way is when POP finds desired label and immediately sends you here with POPOR=[compile-time-case-location of label]. Here we set X=POPOR=[compile-time-case-location]. The other way of arriving here is via receipts. box. and then value of X doesn't matter because PIANO is now

We wish MQ and PPWR thru words to regain original contents before PPOP called. That is purpose of next box. Then JMP + 4  
WTADS you recall is the location you are sent to if instructions is of form "TAD + 5". In this box we first set S=0 to indicate statement has " " rather than label. Next we set X=POPCOR = [C000] location corresponding to " " This is the point where execution starting at line 1

we wish to know what  $n$  is. It is stored in  $M8$  at bits 6-8  
 we move number in to bits 10-12 so right justified and  
 store number at  $R$   
 Now that we know value of  $n$  the next question is whether  
 like "TAD + 5" or "TAD - 5". In next loop which continues on memory  
 box  $\geq = r$ . If negative then  $[bit\ 9] = 1$  and  $\geq = -1$

2  
11

• 418 PALIN V142 13-JUNE-74

(See last page)

Set WORD2 = X + 22 = real location + the label in S1

is S1 refers to.

1437 7654 AND NO SNA CLA  
1448 5249 JMP "4"  
1451 1310 TAD PE  
1452 7141 CIA  
1453 3116 DCA ZL

TAD PE  
TAD X  
DCA HOKU/ADDRESS

Once we have reached this point we know statement is an instruction. Therefore we test to make sure the desired address will

either be on O-page or current page. In this box we take the address stored in WORD2 and make the right 7 bits = 0. If

the address was on page zero this will cause the AC to now = 0

form "JOHN+5" as opposed to "TAD JOHN+5", and you JMP WTADK

otherwise continue

Otherwise we take the left 5 bits of WORD2 and make them = 0. If

AC = 0

1454 1874 TAD HOKU  
1455 1446 AND N7654 SNA CLA  
1456 7454 TAD HOKU  
1457 5314 SNA CIA  
1458 7441 TAD POPCR  
1461 1462 AND N7654 SNA CLA  
1463 7654 TAD HOKU  
1464 5314 SNA CLA  
1465 1432 TAD SSC  
1466 1654 JMP HOKU  
1467 5314 TAD SSC  
1470 4554 JMS 1 OUT  
1471 1414 TAD HOKU  
1472 5522 JMS 1 OUT  
1473 2112 TAD HOKU  
1474 1552 JMS 1 OUT  
1475 8277 TAD HOKU  
1476 4554 TAD HOKU  
1477 5313 TAD HOKU  
1478 1424 TAD HOKU  
1479 1445 TAD HOKU  
1480 1445 TAD HOKU  
1481 1445 TAD HOKU  
1482 1432 TAD HOKU  
1483 5424 TAD HOKU  
1484 1503 TAD HOKU  
1485 1445 TAD HOKU  
1486 1445 TAD HOKU  
1487 1445 TAD HOKU  
1488 1445 TAD HOKU  
1489 1445 TAD HOKU  
1490 1445 TAD HOKU  
1491 1445 TAD HOKU  
1492 1445 TAD HOKU  
1493 1445 TAD HOKU  
1494 1445 TAD HOKU  
1495 1445 TAD HOKU  
1496 1445 TAD HOKU  
1497 1445 TAD HOKU  
1498 1445 TAD HOKU  
1499 1445 TAD HOKU  
1500 1445 TAD HOKU  
1501 1445 TAD HOKU  
1502 1445 TAD HOKU  
1503 1445 TAD HOKU  
1504 1445 TAD HOKU  
1505 1445 TAD HOKU  
1506 1445 TAD HOKU  
1507 1445 TAD HOKU  
1508 1445 TAD HOKU  
1509 1445 TAD HOKU  
1510 1445 TAD HOKU  
1511 1445 TAD HOKU  
1512 1445 TAD HOKU  
1513 1445 TAD HOKU  
1514 1445 TAD HOKU  
1515 1445 TAD HOKU  
1516 1445 TAD HOKU  
1517 1445 TAD HOKU  
1518 1445 TAD HOKU  
1519 1445 TAD HOKU  
1520 1445 TAD HOKU  
1521 1445 TAD HOKU  
1522 1445 TAD HOKU  
1523 1445 TAD HOKU  
1524 1445 TAD HOKU  
1525 1445 TAD HOKU

Now we compare those left 5 bits to the left 5 bits of WORD2. If they are the same, everything is all right, we have a reference to

address on current page and JMP WTADK. Otherwise an error and continue

check to see if SSL#0 indicates user desires to have each compile

here error reported. If this is the case continue. Otherwise JMP WTADK

Increment SSL so that SSL#0 so that you may now print

linefeed > "?" Also increment TADS so you remember SSL was incremented

and N7654. TAD HOKU? This box causes you to calculate the right 8:4 bits associated

with the address of an instruction that refers to current page

You also reach this box when the above compile error occurs

but that doesn't matter since error can not be corrected

CALL HAL This box causes you to calculate the correct left four bits of

instruction. The formula used is [LEFT FOUR BITS] = [BITS 2-5 from MQ]-2

This box causes WORD2 = WORD2+AC = WORD2+LEFT 4 bits = binary number of instructions

Examine bits 2-4 to determine which type of instruction is being handled, then look up instruction table so you can print instruction.

Of course, for example you might print "TAD" yourself.

When this is only done

SSL to

Put bit 5 in AC which is indirect bit that indicates









This is subroutine LLESS; LLES\$. Its purpose is to determine

PAL10 V142 15-JUN-74 1014 PAGE 1-14 the second argument is less than first and if so return to program, skip 2 instruction and the

~11

1761 0414  
1762 4245  
1763 0402  
1764 0147  
1765 1146  
1766 1147  
1767 7141  
1770 1146  
1771 7624  
1772 2361  
1773 5261  
2804 2410  
2805 2411  
2806 2412  
2807 2413  
2808 2414  
2809 2415  
2810 2416  
2811 2417  
2812 2418  
2813 3121  
2814 3114  
2815 1413  
2816 3147  
2817 1146  
2820 7441  
2821 7145  
2822 7214  
2823 5226  
2824 2244  
2825 5645  
2826 1545  
2827 7441  
2834 1507  
2831 7640  
2832 2114  
2833 2102  
2834 2105  
2835 2122  
2836 5226  
2837 1114  
2844 7654  
2841 5254  
2842 5255  
2843 4043  
2844 5252  
2845 5254  
2846 5254  
2847 5256  
2850 1052  
2851 7441

LLESS, JMS I SUBIN word arguments into C and B.

Set AC = -C = two's complement of C and also clear link

TAD C IAC CLU When you TAB B + the link will become = 2 if and only if B ≥ [actual number represented by C]

the actual number represented by C is the octal number represented by B. Thus when B ≥ C you will skip to JMP I LLESS and return on the other hand when B < C you will ISZ LLESS before JMP I LLESS. Thus by

where you return to main-line user will know whether B > C or B < C.

The purpose of subroutine QNAME is to search the area of core between A and B for a particular name. This is done for instance when looking for name of label in symbol table or name of instruction in instruction table. The first box causes arguments to be read in A & B. See Gen Descri.

Set QNAME2 = A = location of next name we are to examine. Set DOLoop = -[left byte of T A] = -[length in words] of this name.

Set W = 0. W will count number of words the name we are examining differs from name we are looking for. Also set C = 2D1SP = [where search-copy of name]

Set SPA CLA JMP I3. Check to see if A has been incremented so much that now A ≥ B. If this is true we have searched whole table and not found desired name, we therefore must return to location of main-line which resolves failures. Therefore we

ISZ QNAME and return via JMP I QNAME. If A < B than there are more items to search on table and JMP +3 to next box.

Compare the A-th word of the name we are examining to C-th word of search-copy. If they do not agree increment W to indicate another item found.

Increment A and C to point to the next words in their respective names. Then ISZ Doloop to see if there are still more words to be compared. If yes go back to QNAME. If no continue

We check to see W = 0. If yes then we have found the name we have found in name we were searching for. Also note that very conveniently, QNAME2=[name's address] consistently we apply return via JMP I QNAME. If W ≠ 0 then we haven't found a name and go back to QNAME where we try again

(4)

THIS

PAGE SAME XEROX AS LAS.

		LLESS, PAGE SOFF HAS104 LIMIT
1761	P40D	JMS 1 SURIN
1762	4545	JMS 2
1763	R4H2	C
1764	R1H7	B
1765	01H6	TAD C
1766	11H7	ISL CLL
1767	7141	TAD H
1770	11H6	SNL CL
1771	762H	ISL LLESS
1772	2361	JMP 1 LLESS
1773	5261	
	P4H0	
2810	R1H9	
2811	4545	QNAME1 A
2812	R4P2	JMS 1 SURIN
2813	R1H2	2
2814	4H2	A
2815	11H6	B
2816	28H5	TAD A
2817	31H5	ISL NAMEZ
2818	3222	UCA
2819	3114	NAME1
2820	4H13	TAD
2821	31H7	AND
2822	11H6	RSN
2823	28H4	RSK6R
2824	15H9	CIA
2825	56H4	UCA
2826	15H9	DISP
2827	7441	TAD
2828	7714	RPA
2829	52H2	JMP
2830	22H4	ISL
2831	764H	NAME
2832	21H4	NAME
2833	15H7	NAME
2834	21H5	NAME
2835	21H2	DOLNP
2836	52H6	JMP
2837	1114	ONAME
2840	76H4	TAD
2841	16H4	SNA
2842	52H5	GLA
2843	4H13	NAME
2844	4H27	JMP
2845	56H4	NAME
2846	52H4	NAME
2847	52H6	JMP
2851	7443	NAME

Sub fun QINST is called by main-line when it is determined that you are passing an instruction - namely QINST discards which type of instruction. See General Descriptions. In the first box you call subroutine QNAME to look up instruction table and report which instruction have. Two instruction table lies between location BEGINS-ENDINS. If you find instruction you JMP +2 to next box. It no such instruction can be found on table return to ERORR since syntax-error occurred.

In the next box which continues thru next page you set OPCODE = (QNAME-BEGINS) ... (with truncation)

When instruction has opcode 0-5 above OPCODE = continuing language opcode + 1.

卷之三

We checked to see if Project 7

6

-2-

四

We check to see if  $OpCode < 7$  indicates we have  
PAGE 1-15 address-instruction. If yes, JMP I NEXT and resume parse.  
If no continue and go to next box.

We wish to make WORD1 = [my standard internal code for mnemonic instruction]. The # indicates which specific mem. Inst. we have in mind. Now in bits 6-12. We shift it 4 left to bits 2-8 and put its 9-12 the octal number 17 which indicates we have mnemonic instruction. The result is stored in WORD1

JMS I NOCHAR WE CALL NOCHAR to parse and make sure next character is carriage return as if should be end of statement

INSTL STA DCA MICRO Set MICRO = MICKR1 = -1 = 7777

INSTL STA DCA MICKR1

The purpose of first box is to locate micro-name on table of micro-instructions. If it can not be found syntax error has occurred and program is stopped.

**AT [T INDEX] = QUAMEZ - MICDPL**

Now **INDEX** is a 0 in every bit where the present name forces instruction to have a 0. Consequently we force MICRO to affect auto-increment in each of these bits. Other bits unchanged.

Set  $\text{NEXTR} = \text{RRIHS}_2$ . Then  $\text{JMP SAVINT}$ , so you know  $\text{RRIHS}_1$  is a 0 in every bit position. We thus similarly

The meanings of MICRO and MICKY will become clearer when we come to pause and initialize next micro-name.

# THIS XEROX is continuation of CLASS

\*482

PAL10

V142

13-JUN-74

16144

PAGE 1-15

2052	1121	TAD DNAME
2053	7114	CLL RAN
2054	3134	DCA OPCODE
2055	1134	AID OPCODE
2056	1431	TAD NEG7
2057	7514	SRA
2060	5264	JMP I NEXT
2061	1464	TAD NEG25
2062	7314	SRA
2063	5273	JMP QINSTR
2064	7100	CLL RTL
2065	7006	RTL
2066	1455	TAD 17
2067	3623	DCA WORD1
2071	4563	JMS I NOWCR
2072	4901	JNS I INSDO
2073	7243	QINSTR, STA 1
2074	3126	DCA MICR
2075	7249	STA MICR
2076	3127	DCA MICR
2077	4684	QINSTR, JMS I NAME
2101	6611	TAD O1 NEE
2101	4633	TAD MICRUP
2102	3415	DCA MICR
2107	1415	TAD INDEX
2110	3126	AND MICR
2111	3126	DCA MICR
2112	1415	TAD I INDEX
2113	6122	AND MICR
2114	3122	DCA MICR
2115	1444	AND MICR
2116	3132	DCA NEXTY
2117	5504	TAD INDEX
2120	7264	QINSTR, CLA
2121	1120	TAD MICR
2122	7421	MOL
2123	3127	TAD MICR
2124	7501	MICR
2125	7001	IAC
2126	7444	SAC
2127	5256	I ERROR
2128	3122	TAD HTR1
2131	7844	CM1
2132	7106	CLL RTL
2133	1332	O I
2134	1332	TAD MICR
2135	3825	CUA HTR1
2136	4563	JMS TNSDO
2137	6494	HICRAT, 2

QINSTR is called by NEXT:NEXT after all the micro-names describing a micro-instruction are read in. We now have 2 tasks  
 1) We determine if the names are compatible with each other i.e. that is they do not require simultaneously a 0 and 1

2) If compatible we calculate binary code of name

See General Description. In the first box we set AC=[MICR0] or [MICR1]. Remember since MICR has a 0 in every bit

is required in and MICR1 has a 1 in every bit of I is required in. Hence the result must be AC=7777 if micro-instruction is compatible

We increment AC. Now it must be =0 if instructions are compatible. We test to see. If  $\neq 0$  JMP I. ERROR since syntax error has occurred. If =0 then compatible. Go to next box

It turns out that the right 4-digits of binary code of micro-instruction can be now calculated by taking complement of  $\text{MICR}_1$ . So we do this

We now have to set WORD1=[my standard internal code for micro instruction]. We do this by moving the crucial bits 4-12 indicating we have have micro instruction two left into 2-10 and put into bits 11-12 the number 2.

We end by calling INSDO which enters WORD1 on list of statements.

This is subroutine QLABEL which is called by NEXT. The purpose is to calculate the number of a label] which = label's address

Page 1 of 8 The Identifiers

Since label not 1" symbol twice we wish to place it there. It's name length is 4 words long. In this box we call G-ARRAG to confirm there is additional space in symbol table. If no space available you will set error message and never return here. If space available you go to next box.

First set `QNAME2 = ENDCON` [label is address]. Then set `ENDCON = ENDCON + Y`. It points to new end of symbol table. Finally exit and return to `NEXT`.

PUOT: CLA [NEXT] DCA DNAMER2 = [label] The purpose is to set  
DCA DNAMER2 is another subfunction called by NEXT. It is called when you parse a period  
NEXT is "TAD.+5". We set DNAMER2=0 to indicate no label. See General Description  
DCA SIGN PDISP is called by NEXT when an address has no displacement as in "TAD A"  
QADINSI CLA TAD DNAMER2 THIS BONE JUST BEFORE PAGE TURN

Purpose of PNDISP is to assist SIGN a default value = -1 to make sign be "+" and ZIN X a default value of 0. Thus the default form of "TAD A" becomes "TAD A+0". See General Description.

See next page for continuation  
of Xerox

74

•400

PALIN	V142	13-JUN-74	16144
2148	4614	DIABEL, JMS 1 DNAME N4000000000000000	PAGE 16144 to calculate the identification number of the label name parsed. In the first box we call subroutine of same name to find that entry label name.
2149	6413	ENDCUN	It will set DNAME = [label-name's address] = [label identifier ID number]
2150	4872	HP 1 NEXT	<del>name on list or not. If name exists then try label name</del>
2151	6568	HP 1	<del>name on list or not. If name exists then try label name</del>
2152	1115	DA 1	<del>name on list or not. If name exists then try label name</del>
2153	3342	JMS 1 GRCAC	<del>name on list or not. If name exists then try label name</del>
2154	4558	WILL N	<del>name on list or not. If name exists then try label name</del>
2155	1472	JN 1 HAVE	<del>name on list or not. If name exists then try label name</del>
2156	1472	PLSP	<del>name on list or not. If name exists then try label name</del>
2157	1115	X	<del>name on list or not. If name exists then try label name</del>
2158	3121	EN CUN	<del>name on list or not. If name exists then try label name</del>
2159	1472	TAD ENCON	<del>name on list or not. If name exists then try label name</del>
2160	1472	UCA ONREF2	<del>name on list or not. If name exists then try label name</del>
2161	3472	TAD INCON	<del>name on list or not. If name exists then try label name</del>
2162	5568	UCA ENDON	<del>name on list or not. If name exists then try label name</del>
2163	1121	PCOLON, CLA	<del>name on list or not. If name exists then try label name</del>
2164	3492	TAD ONAME2	<del>name on list or not. If name exists then try label name</del>
2165	5568	DNP LBL	<del>name on list or not. If name exists then try label name</del>
2166	7204	POOT: CLA	<del>name on list or not. If name exists then try label name</del>
2167	3121	UCA ONONE	<del>name on list or not. If name exists then try label name</del>
2168	5568	WORD2 = DNAME2 = [label identification number]. This has to be done	<del>name on list or not. If name exists then try label name</del>
2169	5568	WORD2 = DNAME2 = [label identification number]. This has to be done	<del>name on list or not. If name exists then try label name</del>
2170	5568	WORD2 = DNAME2 = [label identification number]. This has to be done	<del>name on list or not. If name exists then try label name</del>
2171	7244	END SP, ST	<del>name on list or not. If name exists then try label name</del>
2172	3472	UCA SIGN	<del>name on list or not. If name exists then try label name</del>
2173	3113	DA X	<del>name on list or not. If name exists then try label name</del>
2174	3472	OADMIS, CLA	<del>name on list or not. If name exists then try label name</del>
2175	1121	TAD ONAME2	<del>name on list or not. If name exists then try label name</del>
2176	3472	UCA HORUS2	<del>name on list or not. If name exists then try label name</del>
2177	1122	DNP DIRECT	<del>name on list or not. If name exists then try label name</del>
2178	7108	CLL SCA CLA	<del>name on list or not. If name exists then try label name</del>
2179	7444	TAD OPCODE	<del>name on list or not. If name exists then try label name</del>
2180	7134	RTL	<del>name on list or not. If name exists then try label name</del>
2181	1134	RTL	<del>name on list or not. If name exists then try label name</del>
2182	7134	RTL	<del>name on list or not. If name exists then try label name</del>
2183	1134	RTL OPCODE	<del>name on list or not. If name exists then try label name</del>
2184	7106	RTL	<del>name on list or not. If name exists then try label name</del>
2185	7006	RTL	<del>name on list or not. If name exists then try label name</del>
2186	7106	RTL	<del>name on list or not. If name exists then try label name</del>
2187	7106	RTL	<del>name on list or not. If name exists then try label name</del>
2188	7106	RTL	<del>name on list or not. If name exists then try label name</del>
2189	7106	RTL	<del>name on list or not. If name exists then try label name</del>
2190	7406	RTL	<del>name on list or not. If name exists then try label name</del>
2191	3473	RTL MDRD1	<del>name on list or not. If name exists then try label name</del>
2192	1475	TAD STCR	<del>name on list or not. If name exists then try label name</del>
2193	7504	SMA	<del>name on list or not. If name exists then try label name</del>
2194	7204	CLA	<del>name on list or not. If name exists then try label name</del>
2195	1443	TAD NB	<del>name on list or not. If name exists then try label name</del>
2196	1423	DA MUD1	<del>name on list or not. If name exists then try label name</del>
2197	3423	TAD MUD2	<del>name on list or not. If name exists then try label name</del>
2198	1423	DA MUD2	<del>name on list or not. If name exists then try label name</del>
2199	7644	SEA LLA	<del>name on list or not. If name exists then try label name</del>
2200	1423	TAD N4	<del>name on list or not. If name exists then try label name</del>
2201	1423	DA MUH1	<del>name on list or not. If name exists then try label name</del>
2202	1423	DA MUH2	<del>name on list or not. If name exists then try label name</del>
2203	1423	DA MUH3	<del>name on list or not. If name exists then try label name</del>
2204	3423	SEA CLA	<del>name on list or not. If name exists then try label name</del>
2205	1424	SEA CLA	<del>name on list or not. If name exists then try label name</del>

This is "subfunction" QLABEL which is called by NEXT. In PAGE 16144 to calculate the identification number of the label name parsed. In the first box we call subroutine of same name to find that entry label name. It will set DNAME = [label identifier ID number] = [label identification number]

THIS PAGE IS CONTINUATION OF LAST

The next box which continues to next page cause [0/FULL] to = 1 when WORD2 = 0 and 2 when WORD#0. When 1 we store only WORD1 in list of statements and when 2 we store both WORD1 and WORD2 since we want to remember [label identification number]

(3)

2227	7401	IAC
2230	7401	IAC
2231	3233	DCA : #2
2232	4563	JMS : INSDO
2233	1484	0/ELLIN
2234	7244	OCONN
2235	1115	CLA
2236	7401	TAD Y
2237	3241	IAC
2238	4550	DCA : #2
2241	6408	JMS : GARBAG
2242	1241	BILLIN
2243	4333	TAD : -1
2244	1272	JMS : SHOVE
2245	3244	TAD CUMMAT
2246	2348	DCA : I SHOVEX
2247	4551	ISZ SHOVEX
2250	L813	JMS : MOVE
2251	K113	RDISP
2252	2344	ISZ LINE1
2253	2333	SODINN
2254	2476	ISZ SUMLIN
2255	3133	NOONIN TAD LINE1
2256	3124	TAD DOLLAR
2257	1274	TSF
2260	6441	JMP : -1
2261	5268	JLS
2262	6446	RAD NEWIN
2263	1164	DCA WHERE
2264	3131	TAD STEXT
2265	1271	DCA NEXTY
2266	3152	JMP : NEWIN
2267	5564	DUNLAR 2744
2268	6244	STEXTI STEXT
2271	3425	CUMMATT J
2272	F113	ISZ CUMMATT
2273	F114	ISZ CUMMATT
2274	3151	ISZ CUMMATT
2275	4562	ISZ CUMMATT
2276	3152	ISZ CUMMATT
2277	5255	ISZ CUMMATT
2300	6408	ISZ CUMMATT
2301	4550	ISZ CUMMATT
2302	2344	ISZ CUMMATT
2303	1422	ISZ CUMMATT
2304	6422	ISZ CUMMATT
2305	3412	ISZ CUMMATT
2306	1412	ISZ CUMMATT
2307	1403	ISZ CUMMATT
2310	3423	ISZ CUMMATT
2311	7401	ISZ CUMMATT
2312	1294	ISZ CUMMATT
2313	4333	ISZ CUMMATT
2314	1423	ISZ CUMMATT
2315	3424	ISZ CUMMATT

See last

This is OCONN which is called by NEXTNEXT when we finish parsing a comment and now wish to store it on statement table. See Gen. Descript.

The comment will

occupy Y+1

words on

list

of

statements

and the purpose of

first

box

is to call GARBAG to determine whether there is

space

If no space then you will

not return here and user will

receive error message. It space available so to next box

:

Space. If no space then you will not return here and user will receive error message. It space available so to next box

Call subroutine SHOVE to find the word location you are

to move surrounding information

away so you can store your new statement without destroying

old

Set [I SHOVEX] = COMMENT = 13 = [the word that indicates a comment follows]

Increment SHOVEX to point to next location we will store in and move

main content of comment here

X SHOVEL

PREPARES THE NEXT LINE TO BE INSERTED. You are sent here via

Various Section. If the next line is also the first line among the

sequence of lines now being inserted, enter this section via NODRIN. If

on the other hand you are inserting just another line

among group of lines being inserted, then enter via

SODINN. See gen Description. In the first box we

increment LINE1 so we now point to area where next

line is to be inserted and also increment SUMLIN

which counts total number of lines in text

We set LINE = LINE1 so that LINE-POINTER is updated

We print "g" so user knows he may now type next line to be inserted in text. "g" also tells user he is in text mode

We set WHERE = NEWIN so that program knows where to go after

you begin pure in TEXT-MODE. The JMP NEWIN so

possible error occurs. We set NEXTNEXT = 0 so you can receive next line of typed character,

8460

PAL 16

15-JUN-24

164

This is a continuation

Oct. 1. - P. 1.

八

You reach IIDO when program is in edit-command mode and enter types "I(n)" indicating that he wishes to insert a typed-in text beginning at n-th line. Similarly if user wishes to insert text via printer reader and not have teletype echo text while it is read in type reader and not have teletype echo text while it is read in user types "R(n)" and that causes you to enter this section at RDO. See General description. Since ACT#0 the effect of the first box is to set SSR=0 when "ICn" typed and SSR#0 when "RCn" is typed thus program could distinguish between those cases

Call Decimal to calculate the above line number ( $n$ ) using standard rules. Then deposit that number via LINE1. In the last box we JMP MODR1 so we may enter text mode.

2278	5554	DOLLAR, \$11
2279	8244	TEXT, SIX
2271	3475	SIX
2272	7441	GUMMALL
2273	11001	CLA
2274	3111	DCA SS1
2275	4562	JMS 1 DECIMAL
2276	3133	JMS LINE1
2277	5255	JMP NODR1
234H	6400	INSDO, n
234U	4558	JMS 1 GARBAG
234J	5	J
234S	1422	JAD LABEL
234P	7654	SMA CLA
234L	5312	JMP .TS
230E	1412	TAD NODR1
230D	1423	TAD WDR01
231G	3423	TDC WDR01
231I	7401	IAC
231L	1	LINE1
231J	1	INSDO
231S	TAU	I
231P	JMS	SHUVE
231D	TAU	WDR01
231G	0CA	I SHOVER

In this box we check to see if LABEL=0 indicates statement has no label. If no label JMP.TS to next box. If it has label then set

(37)

448

PALIO

V142

13-JUN-79

1614

PAGE 1-10

1011

1012

1013

1014

1015

1016

1017

1018

1019

1020

1021

1022

1023

1024

1025

1026

1027

1028

1029

1030

1031

1032

1033

1034

1035

1036

1037

1038

1039

1040

1041

1042

1043

1044

1045

1046

1047

1048

1049

1050

1051

1052

1053

1054

1055

1056

1057

1058

1059

1060

1061

1062

1063

1064

1065

1066

1067

1068

1069

1070

1071

1072

1073

1074

1075

1076

1077

1078

1079

1080

1081

1082

1083

1084

1085

1086

1087

1088

1089

1090

1091

1092

1093

1094

1095

1096

1097

1098

1099

1100

1101

1102

1103

1104

1105

1106

1107

1108

1109

1110

1111

1112

1113

1114

1115

1116

1117

1118

1119

1120

1121

1122

1123

1124

1125

1126

1127

1128

1129

1130

1131

1132

1133

1134

1135

1136

1137

1138

1139

1140

1141

1142

1143

1144

1145

1146

1147

1148

1149

1150

1151

1152

1153

1154

1155

1156

1157

1158

1159

1160

1161

1162

1163

1164

1165

1166

1167

1168

1169

1170

1171

1172

1173

1174

1175

1176

1177

1178

1179

1180

1181

1182

1183

1184

1185

1186

1187

1188

1189

1190

1191

1192

1193

1194

1195

1196

1197

1198

1199

1200

1201

1202

1203

1204

1205

1206

1207

1208

1209

1210

1211

1212

1213

1214

1215

1216

1217

1218

1219

1220

1221

1222

1223

1224

1225

1226

1227

1228

1229

1230

1231

1232

1233

1234

1235

1236

1237

1238

1239

1240

1241

1242

1243

1244

1245

1246

1247

1248

1249

1250

1251

1252

1253

1254

1255

1256

1257

1258

1259

1260

1261

1262

1263

1264

1265

1266

1267

1268

1269

1270

1271

1272

1273

1274

1275

1276

1277

1278

1279

1280

1281

1282

1283

1284

1285

1286

1287

1288

1289

1290

1291

1292

1293

1294

1295

1296

1297

1298

1299

1300

1301

1302

1303

1304

1305

1306

1307

1308

1309

1310

1311

1312

1313

1314

1315

1316

1317

1318

1319

1320

• 400	PAL10	PSTAR	CIA	LABEL	PSTAR	1> called by NEXT when parsing "*" as in "1*200" see Gen Descr. Set TAD.
	V142	DCA	13-JUN-24	10:44	because statement has no label. Set SIGN=0 to remember that we PAGE 11 had a "1*200" rather than "1+200" or "-200". Then JMP SNUM to initialize number	
2404	3873	SIGN	JMP SNUM	PSIGN	PSIGN is called by NEXT when parsing "+" and "-" sign in a number to be stored in CORE	
2405	5216	TAD NEG14	01	PSIGN is called by NEXT when parsing "+" and "-" sign in a number to be stored in CORE		
2406	1854	01	01	because statement has no label. Set SIGN=0 to remember that we PAGE 11 had a "1*200" rather than "1+200" or "-200". Then JMP SNUM to initialize number		
2407	3875	01	01	PSIGN is called by NEXT when parsing "+" and "-" sign in a number to be stored in CORE		
2410	5216	01	01	PSIGN is called by NEXT when parsing "+" and "-" sign in a number to be stored in CORE		
2411	7244	01	01	PSIGN is called by NEXT when parsing "+" and "-" sign in a number to be stored in CORE		
2412	1416	01	01	PSIGN is called by NEXT when parsing "+" and "-" sign in a number to be stored in CORE		
2413	3416	01	01	PSIGN is called by NEXT when parsing "+" and "-" sign in a number to be stored in CORE		
2414	7244	01	01	PSIGN is called by NEXT when parsing "+" and "-" sign in a number to be stored in CORE		
2415	3875	01	01	PSIGN is called by NEXT when parsing "+" and "-" sign in a number to be stored in CORE		
2416	1855	01	01	PSIGN is called by NEXT when parsing "+" and "-" sign in a number to be stored in CORE		
2417	3113	01	01	PSIGN is called by NEXT when parsing "+" and "-" sign in a number to be stored in CORE		
2420	3115	01	01	PSIGN is called by NEXT when parsing "+" and "-" sign in a number to be stored in CORE		
2421	5504	01	01	PSIGN is called by NEXT when parsing "+" and "-" sign in a number to be stored in CORE		
2422	1456	01	01	PSIGN is called by NEXT when parsing "+" and "-" sign in a number to be stored in CORE		
2423	3116	01	01	PSIGN is called by NEXT when parsing "+" and "-" sign in a number to be stored in CORE		
2424	1115	01	01	PSIGN is called by NEXT when parsing "+" and "-" sign in a number to be stored in CORE		
2425	7106	01	01	PSIGN is called by NEXT when parsing "+" and "-" sign in a number to be stored in CORE		
2426	7494	01	01	PSIGN is called by NEXT when parsing "+" and "-" sign in a number to be stored in CORE		
2427	1116	01	01	PSIGN is called by NEXT when parsing "+" and "-" sign in a number to be stored in CORE		
2430	3115	01	01	PSIGN is called by NEXT when parsing "+" and "-" sign in a number to be stored in CORE		
2431	2113	01	01	PSIGN is called by NEXT when parsing "+" and "-" sign in a number to be stored in CORE		
2432	5504	01	01	PSIGN is called by NEXT when parsing "+" and "-" sign in a number to be stored in CORE		
2433	5536	01	01	PSIGN is called by NEXT when parsing "+" and "-" sign in a number to be stored in CORE		
2434	7284	01	01	PSIGN is called by NEXT when parsing "+" and "-" sign in a number to be stored in CORE		
2436	3134	01	01	PSIGN is called by NEXT when parsing "+" and "-" sign in a number to be stored in CORE		
2437	3125	01	01	PSIGN is called by NEXT when parsing "+" and "-" sign in a number to be stored in CORE		
2440	5564	01	01	PSIGN is called by NEXT when parsing "+" and "-" sign in a number to be stored in CORE		
2441	7284	01	01	PSIGN is called by NEXT when parsing "+" and "-" sign in a number to be stored in CORE		
2442	3125	01	01	PSIGN is called by NEXT when parsing "+" and "-" sign in a number to be stored in CORE		
2443	5564	01	01	PSIGN is called by NEXT when parsing "+" and "-" sign in a number to be stored in CORE		
2444	4540	01	01	PSIGN is called by NEXT when parsing "+" and "-" sign in a number to be stored in CORE		
2445	6413	01	01	PSIGN is called by NEXT when parsing "+" and "-" sign in a number to be stored in CORE		
2446	8115	01	01	PSIGN is called by NEXT when parsing "+" and "-" sign in a number to be stored in CORE		
2447	2105	01	01	PSIGN is called by NEXT when parsing "+" and "-" sign in a number to be stored in CORE		
2448	1475	01	01	PSIGN is called by NEXT when parsing "+" and "-" sign in a number to be stored in CORE		
2451	7454	01	01	PSIGN is called by NEXT when parsing "+" and "-" sign in a number to be stored in CORE		
2453	7184	01	01	PSIGN is called by NEXT when parsing "+" and "-" sign in a number to be stored in CORE		
2454	7124	01	01	PSIGN is called by NEXT when parsing "+" and "-" sign in a number to be stored in CORE		
2455	7124	01	01	PSIGN is called by NEXT when parsing "+" and "-" sign in a number to be stored in CORE		
2456	1115	01	01	PSIGN is called by NEXT when parsing "+" and "-" sign in a number to be stored in CORE		
2457	7006	01	01	PSIGN is called by NEXT when parsing "+" and "-" sign in a number to be stored in CORE		
2458	7494	01	01	PSIGN is called by NEXT when parsing "+" and "-" sign in a number to be stored in CORE		
2461	7001	01	01	PSIGN is called by NEXT when parsing "+" and "-" sign in a number to be stored in CORE		
2462	3823	01	01	PSIGN is called by NEXT when parsing "+" and "-" sign in a number to be stored in CORE		
2463	4563	01	01	PSIGN is called by NEXT when parsing "+" and "-" sign in a number to be stored in CORE		
2464	1115	01	01	PSIGN is called by NEXT when parsing "+" and "-" sign in a number to be stored in CORE		
2465	3824	01	01	PSIGN is called by NEXT when parsing "+" and "-" sign in a number to be stored in CORE		
2466	1475	01	01	PSIGN is called by NEXT when parsing "+" and "-" sign in a number to be stored in CORE		
2467	7454	01	01	PSIGN is called by NEXT when parsing "+" and "-" sign in a number to be stored in CORE		
2468	7454	01	01	PSIGN is called by NEXT when parsing "+" and "-" sign in a number to be stored in CORE		
2471	5275	01	01	PSIGN is called by NEXT when parsing "+" and "-" sign in a number to be stored in CORE		

\*408 PAL10 V142 14-JUN-74 16144 PAGE 1-10 CONTINUATION OF XEROX

2404	38779	DCA SIGN
2405	52110	JMP SNUM
2406	1454	TAD NLG14
2407	38775	UCA SIGN
2410	52116	JMP SNUM
2411	7224	STA
2412	14116	TAD NLXCAH
2413	54116	DCA MEXCAN
2414	7224	STA
2415	38779	DCA SIGN
2416	1454	JMP TAD NLG5
2417	3113	SNUM
2418	3113	DCA X
2421	5554	DCA SIGN
2422	1456	JMP TAD NEG27
2423	3116	PNUM
2424	3115	DCA X
2425	7116	TAD Y
2426	7116	CLL NL
2427	1110	TAD HAL
2430	3115	DCA X
2431	7113	TAD Y
2432	5264	IMP X
2433	5260	PATSN1
2434	7224	NATSH
2435	3110	DCA OUTDUE
2436	7240	PATSN2
2437	3115	ONUM:
2440	5260	UCA DIRECT
2441	7214	JMP I NEXT
2442	3125	CLA DIRECT
2443	5264	JMP I NEXT
2444	4546	HAL
2445	5213	HAL
2446	7219	HAL
2447	1262	JMP ONUML
2450	1073	TAD SIGN
2451	7454	SNA
2452	5265	HAL ONUML
2453	7116	CLL SNA
2454	7214	CLA SNA
2455	7116	STA
2456	1110	TAD
2457	7116	RTL
2459	7116	HAL
2460	7116	IAC
2461	7461	DCA HORUS
2462	31023	JMS TRSDO
2463	4563	HAL
2464	1073	DCA HORUS?
2465	1335	ONUML:
2466	3424	TAD SICH
2467	1472	SNA
2470	7454	HAL ONUML
2471	5275	STA CLA
TAD	M20	TAD, N4

QNUML is called by NEXT after you have finished parsing the 3 forms: " + 127", " - 7172" or " 7172" To purpose of QNUML is to store in the list of statements information. See Gen Descr. In the first box we test to see if this information is stored in TAD box. If it is, we have absolute value less than 400. If  $\geq 400$  JMP QNUML. Otherwise continue.

If  $SIGN = 0$  indicating presence of "\*" we JMP QNUML in this case. If  $SIGN = -1$  indicating "+" and  $LINK = 1$  if  $AC = SIGN = 1$  indicating "-".

By the time we leave this box  $AC = 0$ . Also  $LINK = 0$  if too. Otherwise go to next box with  $AC = SIGN = 1$ .

In this box we put in WORD1 the information that is to be stored in list of statements. Crucial to understanding what I am doing here is that  $Y \leq 400$  and the number is therefore being stored in short-number notation. We begin this box by putting object with the link. Set appropriately 3 left in AC. Finally we put "01" in bits 11-12 of +ve AC to indicate we have a number in short notation. Then we store result in WORD1.

We exit via INS0 which puts information on list of statements.

At ONUML we prepare long positive and negative numbers and also \*-numbers in long notation. In first box we set WORD2 =  $=$  absolute value of number.

If  $SIGN = 0$  indicating \* JMP QNUML

In this box which continues to next page we set [bit 8] = 1 if  $SIGN = 1$  indicating "+" and set [bit 8] = 0 if  $SIGN = -1$  indicating "-" whether "+" or "-" we set [bit 10] = 1 to indicate we have number rather than "\*".

•400

PAL10

V142

L-J-JUN-  
LAST

B4-E

6:09

PAGE 1-20

proper

in

WORKD1

. Finally we exit via

INSDO

which stores info. on list-store.

110

0:09

Subroutine

DECIMAL

DDECM

is called when you decipher user's type is decimal

number and use

pointers to evaluate which lines he is referring to

In first box we initialize the number

when he typed in

last command

when he typed in

number and use

INSDO

when he typed in

This is a short LOVE: LOVE is control by DDDOMIND, LCDT and others. It's a  
line-number and study their relation. See Gen Descr. In first bo.  
15-JUN-24 1014 PAGE 1-21 call DECIMAL to calculate first line number and deposit result

LC21 Line 21  
 JHS I LESS  
 LINE1  
 LINE2  
 JMP I ERROR  
 TAD LC1  
 CIA  
 TAD LC2  
 DCA H  
 TAD LINE1

Set line-pointer = LINE + 0 = LINE1. Then call subr NOWCAR to make sure next char is <Carriage ret>. If no <Car rot>, NOWCAR causes error message. If yes, then NOWCAR returns control to this program which then JMP I LOVE so you return to program that called LOVE

DDDD is a subfunction of NEXT. It causes you to delete all the lines in stored

We set  $X = BEGWORK$  and  $BEGWORK = BEGWORK + w$  [where list of statements will begin after deletion]. This is where actual deletion occurs. We move information stored between  $L_{i-1}$  and  $L_i$  to starting location  $BEGWORK$ . Locations  $X$  thru  $L_{i-1}$  are deleted. The first time total number of lines has been decreased by  $i$ .

We make SUMLIN reflect +20 feet.  
 $L_1 = SUMLIN - LINE1 + LINE2$

decreased by setting  $SUMLIN = SUMLIN + LINE1 - LINE2$

JMP I WHERE causes us to return to EDIT and receive next EDIT COMMAND

MENDO is a subfunction of next. It is called when user wishes to "move" lines

Specifically MENDO reads 3 line numbers  $n_1, n_2$ , and  $n_3$  and just before line

it puts a duplicate copy of lines  $n_2$  thru  $n_3$  inclusive. See General Description

In the Box call DECML to calculate first line number and store it in LINE

FIND-POPUP. In this box which continues on next page we call FIND. FIND sets

~~DATA MANDOX~~ ~~MANDOX = [ address of first word of LINE ]. ALSO FIND sets FORMER=MANDOX - I: because we want  
unusual value we reset MANDOX=POPULAR~~

• 400

PALIO

VIAZ

13-JUN-24

1614

TAO POPWOR

DCA MMDOX

JHS LOVE

TAD W

ICA \*2

ICA

JHS GARBAR

GARBAR

W/FILLIN

SIA

TAU LC2

DCA LC2

JHS MOVE

LCL

L2

ENDCON

ENDCON

TAD C

DCA H

MMDOLL TAD H

CIA H

TAD REGOR

DCA X

STA H

TAD BEGMR

DCA BEGMR

JHS MOVE

X MMDOK

HIGHOR

TAO I H

DCA H

MMDOLL

TAO I H

DCA H

14

PALP	Y142	15-JUN-24
2756	7201	LLDDI
2757	3104	CLA IAC
2760	7413	SSC will = 0
2761	3103	+ the same way
2762	3862	SSA was = 0 if you didn't go thru MENDO
2763	2135	
2764	4562	
2765	3077	
2766	3135	
2767	4284	
2768	4556	
2770	1677	
2772	0434	
2773	7112	
2774	7112	
2775	7112	
2776	7118	
2777	3477	
3080	1133	
3081	7491	
3082	1134	
3083	3134	
3084	3134	
3085	7640	
3086	5213	
3087	4551	
3088	7412	
3089	7472	
3090	9124	
3091	3922	
3092	7492	
3093	4543	
3094	8133	
3095	3133	
3096	3133	
3097	3133	
3098	3133	
3099	3133	
3100	3133	
3101	3133	
3102	3133	
3103	3133	
3104	3133	
3105	3133	
3106	3133	
3107	3133	
3108	3133	
3109	3133	
3110	3133	
3111	3133	
3112	3133	
3113	3133	
3114	3133	
3115	4543	
3116	3133	
3117	3133	
3118	3133	
3119	3133	
3120	3133	
3121	3133	
3122	3133	
3123	3133	
3124	3133	
3125	3133	
3126	3133	
3127	3133	
3128	3133	
3129	3133	
3130	3133	
3131	3133	
3132	3133	
3133	3133	
3134	3133	
3135	3133	
3136	3133	
3137	3133	
3138	3133	
3139	3133	
3140	3133	
3141	3133	
3142	3133	
3143	3133	
3144	3133	
3145	3133	
3146	3133	
3147	3133	
3148	3133	
3149	3133	
3150	3133	
3151	3133	
3152	3133	
3153	3133	
3154	3133	
3155	3133	
3156	3133	
3157	3133	
3158	3133	
3159	3133	
3160	3133	
3161	3133	
3162	3133	
3163	3133	
3164	3133	
3165	3133	
3166	3133	
3167	3133	
3168	3133	
3169	3133	
3170	3133	
3171	3133	
3172	3133	
3173	3133	
3174	3133	
3175	3133	
3176	3133	
3177	3133	
3178	3133	
3179	3133	
3180	3133	
3181	3133	
3182	3133	
3183	3133	
3184	3133	
3185	3133	
3186	3133	
3187	3133	
3188	3133	
3189	3133	
3190	3133	
3191	3133	
3192	3133	
3193	3133	
3194	3133	
3195	3133	
3196	3133	
3197	3133	
3198	3133	
3199	3133	
3200	3133	

LLDDI is reached either after MENDO on last punch or directly. This is a subroutine from NEXT after L-command. In box we set SSC=1 to " ". In former case SSC=AC=0 which actually was its previous value and has no real effect. In latter case SSC=AC#0 and thus we have a punch knowing we entered thru BBDO and binary-type but unnameable.

SKP IS  
humless but unnameable

LLDDI  
BHD01 DCA SSBR  
L2D01 DCA CORDUM  
JMS 1 DECMAL  
DCA SSBR  
JMS 1 SSCL  
JMS LOVE  
JMS 1 NOWCAR  
TAD CORDUL  
TAD LINE1  
CIA LINE2  
TAD LINE2  
TAD HOLD  
SEA CLA  
JMP \*5  
JMS 1 MOVE  
HEGCON  
ENDCON  
LIMIT  
DCA LABEL  
HLT  
JMS 1 POP  
LINE1  
BEGWR  
POPD1  
JMS TRAIL THIS DONE AS PAGE TURN CHARACTER. If YES NOWCAR WILL PULL YOU TO  
LINK PUPDR  
PUPDR  
TAU CORDUM  
NSW  
AND MASK6H  
JMS 1 OUTB  
TAU CORDUM  
AND MASKFH  
JMS OUTB  
JMS TRAIL  
JMS SHIFT  
JMP 1 SSCL  
SHIFT  
TAU  
TAD NEG25  
TAD DOLDR  
TAD N20B

In this box we put right 5-digits of CORDPL in AC and reset CORDPL=2. AC. This is because I decided to only allow relocation by punches and because I concluded it would be easiest to use it if we save me right 5-digits.

For rest of this page see next box

Set LINE2 = LINE2 - LINE1 because we want to process a total number of (LINE2 - LINE1) lines.

Set LABEL=0 since we will not be searching for any label when we call subroutine POP. Then HLT so user can turn on tape punch if he desires.

The second time we call POP we have different arguments. The reason for this call is that POPWORK is an argument of second time POP called. We call TRAIL for user's convenience and print trailing for possible tape output according to values of SSN, SSL, SSR, SSC.

When SSN#0, output causes us to punch first two left bytes and then the rest of check sum. This is to conform to standard of DEC-BIN reader. Call TRAIL to print trailers. Call SHIFTSWATCH to put binary on O page. Then JMS I START when you return to initial WAIT mode. Argument LINE2 causes next LINE2 lines to be processed. Finally POP2 causes output activity to be produced

When SSN#0, output causes us to punch first two left bytes and then the rest of check sum. This is to conform to standard of DEC-BIN reader. Call TRAIL to print trailers. Call SHIFTSWATCH to put binary on O page. Then JMS I START when you return to initial WAIT mode. Argument LINE2 causes next LINE2 lines to be produced

This is subr. TRAIL which purpose is to print 256 trailing code of TRAILING

DOLoop = -2.5 and AC = 2.00

46

INSERT  
ISNUXME  
JMP. -5

3445	6441
3446	5245
3447	64400
3450	56400
3451	72000
3452	31022
3453	3053
3454	1412
3455	74411
3456	16722
3457	14711
3458	31241
3461	12124
3462	45533
3463	31021
3464	31011
3465	31131
3466	55666
3467	45544
3468	31021
3469	55331
3470	24444
3472	3072
3473	45455
3474	42444
3475	210995
3476	31021
3477	21077
3478	21066
3479	31021
3480	454000
3481	31021
3482	61000
3483	454000
3484	53334
3485	11165
3486	74411
3487	21077
3488	21066
3489	31021
3490	454000
3491	31021
3492	61000
3493	454000
3494	53334
3495	11165
3496	74411
3497	21077
3498	21066
3499	31021
3500	454000
3501	31021
3502	61000
3503	454000
3504	53334
3505	11165
3506	74411
3507	21077
3508	21066
3509	31021
3510	454000
3511	31021
3512	61000
3513	454000
3514	53334
3515	11165
3516	74411
3517	21077
3518	21066
3519	31021
3520	454000
3521	31021
3522	61000
3523	454000
3524	53334
3525	11165
3526	74411
3527	21077
3528	21066
3529	31021
3530	454000
3531	31021
3532	61000
3533	454000
3534	53334
3535	11165
3536	74411
3537	21077
3538	21066
3539	31021
3540	454000
3541	31021
3542	61000
3543	454000
3544	53334
3545	11165
3546	74411
3547	21077
3548	21066
3549	31021
3550	454000
3551	31021
3552	61000
3553	454000
3554	53334
3555	11165
3556	74411
3557	21077
3558	21066
3559	31021
3560	454000
3561	31021
3562	61000
3563	454000
3564	53334
3565	11165
3566	74411
3567	21077
3568	21066
3569	31021
3570	454000
3571	31021
3572	61000
3573	454000
3574	53334
3575	11165
3576	74411
3577	21077
3578	21066
3579	31021
3580	454000
3581	31021
3582	61000
3583	454000
3584	53334
3585	11165
3586	74411
3587	21077
3588	21066
3589	31021
3590	454000
3591	31021
3592	61000
3593	454000
3594	53334
3595	11165
3596	74411
3597	21077
3598	21066
3599	31021
3600	454000
3601	31021
3602	61000
3603	454000
3604	53334
3605	11165
3606	74411
3607	21077
3608	21066
3609	31021
3610	454000
3611	31021
3612	61000
3613	454000
3614	53334
3615	11165
3616	74411
3617	21077
3618	21066
3619	31021
3620	454000
3621	31021
3622	61000
3623	454000
3624	53334
3625	11165
3626	74411
3627	21077
3628	21066
3629	31021
3630	454000
3631	31021
3632	61000
3633	454000
3634	53334
3635	11165
3636	74411
3637	21077
3638	21066
3639	31021
3640	454000
3641	31021
3642	61000
3643	454000
3644	53334
3645	11165
3646	74411
3647	21077
3648	21066
3649	31021
3650	454000
3651	31021
3652	61000
3653	454000
3654	53334
3655	11165
3656	74411
3657	21077
3658	21066
3659	31021
3660	454000
3661	31021
3662	61000
3663	454000
3664	53334
3665	11165
3666	74411
3667	21077
3668	21066
3669	31021
3670	454000
3671	31021
3672	61000
3673	454000
3674	53334
3675	11165
3676	74411
3677	21077
3678	21066
3679	31021
3680	454000
3681	31021
3682	61000
3683	454000
3684	53334
3685	11165
3686	74411
3687	21077
3688	21066
3689	31021
3690	454000
3691	31021
3692	61000
3693	454000
3694	53334
3695	11165
3696	74411
3697	21077
3698	21066
3699	31021
3700	454000
3701	31021
3702	61000
3703	454000
3704	53334
3705	11165
3706	74411
3707	21077
3708	21066
3709	31021
3710	454000
3711	31021
3712	61000
3713	454000
3714	53334
3715	11165
3716	74411
3717	21077
3718	21066
3719	31021
3720	454000
3721	31021
3722	61000
3723	454000
3724	53334
3725	11165
3726	74411
3727	21077
3728	21066
3729	31021
3730	454000
3731	31021
3732	61000
3733	454000
3734	53334
3735	11165
3736	74411
3737	21077
3738	21066
3739	31021
3740	454000
3741	31021
3742	61000
3743	454000
3744	53334
3745	11165
3746	74411
3747	21077
3748	21066
3749	31021
3750	454000
3751	31021
3752	61000
3753	454000
3754	53334
3755	11165
3756	74411
3757	21077
3758	21066
3759	31021
3760	454000
3761	31021
3762	61000
3763	454000
3764	53334
3765	11165
3766	74411
3767	21077
3768	21066
3769	31021
3770	454000
3771	31021
3772	61000
3773	454000
3774	53334
3775	11165
3776	74411
3777	21077
3778	21066
3779	31021
3780	454000
3781	31021
3782	61000
3783	454000
3784	53334
3785	11165
3786	74411
3787	21077
3788	21066
3789	31021
3790	454000
3791	31021
3792	61000
3793	454000
3794	53334
3795	11165
3796	74411
3797	21077
3798	21066
3799	31021
3800	454000
3801	31021
3802	61000
3803	454000
3804	53334
3805	11165
3806	74411
3807	21077
3808	21066
3809	31021
3810	454000
3811	31021
3812	61000
3813	454000
3814	53334
3815	11165
3816	74411
3817	21077
3818	21066
3819	31021
3820	454000
3821	31021
3822	61000
3823	454000
3824	53334
3825	11165
3826	74411
3827	21077
3828	21066
3829	31021
3830	454000
3831	31021
3832	61000
3833	454000
3834	53334
3835	11165

45

PAL10

4142

13-JUN-74

1014 PROT-1

D

A = B indicates no more information is to be transferred. If A = B exit from subroutine via JMP I MOVE S2A CLA which causes you to return to calling program. If A ≠ B, more JMS MOVIE information should be transferred and go back to MOVE

3154	1145	TAD A
3155	1241	Test + to see if be transferred.
3156	106	If A = B exit from subroutine via JMP I MOVE
3157	764	S2A CLA which causes you to return to calling program. If A ≠ B, more
3140	5338	JMP MOVIE information should be transferred and go back to MOVE
3141	5672	JMS
3142	3343	POPO1, POPD1
3143	1121	PUP011, POPKL
3144	1121	POPKL
3145	1121	POP2N
3146	1123	POPKL
3147	1121	POPKL
3150	1121	POPKL
3151	1121	POPKL
3152	1124	POPKL
3153	1121	POPKL
3154	1121	POPKL
3155	1121	POPKL
3156	1123	POPKL
3157	1121	POPKL
3164	1121	POPKL
3161	1121	POPKL
3162	1121	POPKL
3163	1164	POPO2, POPD2
3164	1373	PUP022,
3165	1331	WTADL
3166	1262	WPSHN
3167	2464	WHICRO
3178	1424	POST
3171	1338	WTADS
3172	1262	WNSHN
3173	1337	WHICRO
3174	1373	WTADL
3175	1331	WPSHN
3176	1262	WHICRO
3177	1226	WCMM
3208	1424	WTADS
3281	1338	WNSHN
3202	1262	WHICRO
3243	1241	WNE
3274	1262	WHICRO
3284	WTADL	PPWUR, POPOR
3285	1323	WWORD1, WORD1
3286	7578	TEMP1, TEMP6, 7758
3287	7256	Bobbins, 7756
3214	Paru	
3211	PPWUR	
3212	1341/AND	
3213	5644	
3214	6364	
3215	4144	
3216	4351/158	
3217	6372	
3274	6349/106	
3221	4341	
3222	6352/JMS	

Pointers  
to subfunctions  
of POPS

List of  
Instruction  
Names

(17)

PAL10 V142 1J-JUN-74 16144 PAGE 1-26

3223	5563	5263
3224	6352	0352/JMP
3225	5568	5268
3226	6363	B89H1C1#363/SHM
3227	5241	5241
3228	6363	6363/S2A
3229	7241	7241
3231	0363	0363/SNL
3232	5654	5654
3233	6363	6363/SPA
3234	6041	6041
3235	6363	6363/S2L
3236	0363	0363/SK#
3237	5364	5364
3240	6363	0363/SNA
3241	5641	5041
3247	6362	6362
3249	6359	0359/HLI
3249	7254	7254
3244	0343	0343/CLA
3245	5441	5441
3246	7357	0357/OSK
3247	6362	6362
3249	6359	0359/HLI
3251	5464	5464
3252	0343	0343/CLL
3253	5454	5454
3254	6351	0351/IAC
3255	4143	4143
3256	0343	0343/CMA
3257	5241	5241
3264	6343	6343/CML
3261	5254	5254
3262	6362	6362/RAR
3263	4162	4162
3264	6362	0362/RTH
3265	6462	6462
3266	6362	6362/RAL
3267	4154	4154
3270	6362	0362/HTL
3271	6454	6454
3272	0342	0342/USM
3273	6362	6362
3274	6355	0355/MOA
3275	6141	6141
3276	6355	0355/MOL
3277	6154	6154
3278	0356	ELMIC.0356/MOP
3301	5764	5764
3302	6343	0343/CIA
3303	5141	5141
3304	6363	0363/STL
3305	6454	6454
3306	6363	0363/STA
3307	6441	6441
3310	6342	0342/GLB
3311	5453	5453

Continued  
from last  
page

PAL10 V142

13-JUN-74

16144 PAGE 1-27

3312	8363	0363/SNP
3313	6764	676M
3314	6343	0343/CAM
3315	4155	4155
3316	6351	0351/TDN
3317	5256	5256
3320	8351	0351/10E
3321	5246	5246
3322	8363	0363/SRQ
3323	6162	6162
3324	8342	0342/6TF
3325	6946	6946
3326	8361	0361/RTE
3327	6446	6446
3328	8363	0363/SGT
3330	6346	6346
3331	4264	4264
3332	8343	0343/SAP
3333	4146	4146
3334	8353	0353/KCF
3335	4346	4346
3336	8353	0353/KSE
3337	6346	6346
3340	8353	0353/KCC
3341	4343	4343
3342	6353	0353/KRS
3343	6163	6163
3344	8353	0353/KIE
3345	5145	5145
3346	8353	0353/KRB
3347	6142	6142
3348	8364	0364/IFL
3351	4654	4654
3352	8364	0364/TSE
3353	6346	6346
3354	8364	0364/TCF
3355	4346	4346
3356	8364	0364/10C
3357	6843	6843
3360	8364	0364/TSM
3361	6353	6353
3362	8364	0364/TLS
3363	5463	5463
3364	8463	8463/SKON
3365	5352	5352
3366	5608	5608

END OF  
LIST OF  
INSTRUCTIONS  
NAMES

DESCRIPTIONS

ELETRINS,  
BuHMCD 2766/SMA

LIST OF  
MICRO INSTRUCTIONS

•448

PAL10 V142

13-JUN-74

16144 PAGE 1-28

4.1

3400	7362	7367
3401	7776	7776/SNA
3402	7322	7327
3403	7276	7776/S2L
3404	7347	7347
3405	7277	7777/CLA
3406	7577	7577
3407	7277	7777/USR
3410	7375	7375
3411	7777	7777/HLT
3412	7376	7376
3413	7377	7377/CLL
3414	7677	7677
3415	7377	7377/AAC
3416	7776	7776
3417	7377	7377/CMA
3420	7232	7232
3421	7377	7377/CMU
3422	7252	7252
3423	7375	7375/HAR
3424	7262	7262
3425	7377	7377/RTR
3426	7265	7265
3427	7375	7375/HAL
3430	7273	7273
3431	7377	7377/RTL
3432	7271	7271
3433	7263	7263/PUSH
3434	7275	7275/PUSH
3435	7272	7777/MOA
3436	7276	7276
3437	7272	7777/MOL
3440	7356	7356
3441	72884	KLK3:0
3443	7441	RUMNEC 2000/NOP
3443	7241/CLA	
3444	7120/S1L	
3445	7240/STA	
3446	7204/GUR	
3447	7521/SHP	
3450	7621/TAM	
3451	6001/ON	
3452	6002/10F	
3453	6003/SRU	
3454	6004/GTE	
3455	6005/KTF	
3456	6006/SGT	
3457	6007/CAF	
3460	6030/KCF	
3461	6031/SE	
3462	6032/KCC	
3463	6034/KRS	
3464	6035/KIE	
3465	6036/KRS	
3466	6040/TPL	

## END LIST

OF  
MICRO-INSTRUCTIONS  
DESCRIPTION CODE

LIST OF  
BINARY CODE  
OP MNEMONIC  
INSTRUCTIONS

•488

PALIO V142

13-JUN-24

16144 PAGE 1-29

New Table

3467 6841 6841/TSE  
 3470 6842 6842/TCE  
 3471 6844 6844/TPC  
 3472 6845 6845/TSK  
 3473 6846 6846/TLS  
 3474 6848 6848/SKD

STTT, 1  
 7241/2A  
 NAME  
 RLABEL

3475 6801 8888  
 3476 7640 8888  
 3477 6821 8888  
 3501 6833 8888  
 3502 6812 8888  
 3504 2482 8888  
 3505 6812 8888  
 3506 1577 8888  
 3507 3531 8888  
 3510 7241 8888  
 3511 1577 8888  
 3512 3513 8888  
 RLABEL, 3  
 3514 6820 8888  
 3515 2149 8888  
 3516 3225 8888  
 3517 7241 8888  
 3520 6704 8888  
 3521 3513 8888  
 3522 3124 8888  
 3523 6268 8888  
 3524 3513 8888  
 3525 6881 8888  
 3526 6814 8888  
 3527 2162 8888  
 3528 3542 8888  
 RCOMM, 2  
 3531 6802 8888  
 3532 7689 8888  
 3533 6268 8888  
 3534 3531 8888  
 3535 6872 8888  
 3536 2234 8888  
 3537 3475 8888  
 3540 6802 8888  
 3541 6804 8888  
 3542 2043 8888  
 3543 3661 8888  
 3544 7241 8888  
 3545 6268 8888  
 3546 3544 8888  
 3547 6805 8888  
 3550 6813 8888  
 3551 2480 8888  
 3552 3642 8888  
 3553 6815 8888

PARS  
 TABLES  
 On next  
 few pages

3478 6800/ ANYTHING  
 3479 7241/2A  
 3480 6800/ ANYTHING  
 3481 6800/ ANYTHING  
 3482 6800/ ANYTHING  
 3483 6800/ ANYTHING  
 3484 6800/ ANYTHING  
 3485 6800/ ANYTHING  
 3486 6800/ ANYTHING  
 3487 6800/ ANYTHING  
 3488 6800/ ANYTHING  
 3489 6800/ ANYTHING  
 3490 6800/ ANYTHING  
 3491 6800/ ANYTHING  
 3492 6800/ ANYTHING  
 3493 6800/ ANYTHING  
 3494 6800/ ANYTHING  
 3495 6800/ ANYTHING  
 3496 6800/ ANYTHING  
 3497 6800/ ANYTHING  
 3498 6800/ ANYTHING  
 3499 6800/ ANYTHING  
 3500 6800/ ANYTHING  
 3501 6800/ ANYTHING  
 3502 6800/ ANYTHING  
 3503 6800/ ANYTHING  
 3504 6800/ ANYTHING  
 3505 6800/ ANYTHING  
 3506 6800/ ANYTHING  
 3507 6800/ ANYTHING  
 3508 6800/ ANYTHING  
 3509 6800/ ANYTHING  
 3510 6800/ ANYTHING  
 3511 6800/ ANYTHING  
 3512 6800/ ANYTHING  
 3513 6800/ ANYTHING  
 3514 6800/ ANYTHING  
 3515 6800/ ANYTHING  
 3516 6800/ ANYTHING  
 3517 6800/ ANYTHING  
 3518 6800/ ANYTHING  
 3519 6800/ ANYTHING  
 3520 6800/ ANYTHING  
 3521 6800/ ANYTHING  
 3522 6800/ ANYTHING  
 3523 6800/ ANYTHING  
 3524 6800/ ANYTHING  
 3525 6800/ ANYTHING  
 3526 6800/ ANYTHING  
 3527 6800/ ANYTHING  
 3528 6800/ ANYTHING  
 3529 6800/ ANYTHING  
 3530 6800/ ANYTHING  
 3531 6800/ ANYTHING  
 3532 6800/ ANYTHING  
 3533 6800/ ANYTHING  
 3534 6800/ ANYTHING  
 3535 6800/ ANYTHING  
 3536 6800/ ANYTHING  
 3537 6800/ ANYTHING  
 3538 6800/ ANYTHING  
 3539 6800/ ANYTHING  
 3540 6800/ ANYTHING  
 3541 6800/ ANYTHING  
 3542 6800/ ANYTHING  
 3543 6800/ ANYTHING  
 3544 6800/ ANYTHING  
 3545 6800/ ANYTHING  
 3546 6800/ ANYTHING  
 3547 6800/ ANYTHING  
 3548 6800/ ANYTHING  
 3549 6800/ ANYTHING  
 3550 6800/ ANYTHING  
 3551 6800/ ANYTHING  
 3552 6800/ ANYTHING  
 3553 6800/ ANYTHING

\*489

PAL10 V142

13-JUN-74

16144 PAGE 1-30

51

3554	2400	PSIGN
3555	3647	RNUM
3556	3129	312670
3557	2411	PNUMB
3560	3647	RNUH
3561	7241	72412A
3562	1577	SNAME
3563	3540	RINST
3564	2940	48/9
3565	2434	PATSN1
3566	3610	RADRES
3567	8403	RINST2,
3570	8080	0/BLANK
3571	2872	OINSTM
3572	3562	RINST2
3573	7241	7241/2A
3574	0760	PNAME
3575	3562	RINST2
3576	6877	77/CAR
3577	2120	OINSTF
3604	3475	SIEXT
3601	6402	RATSN,
3602	8804	0/BLANK
3603	2441	NATSN
3604	3610	RADRES
3605	2248	48/9
3606	2436	PATSN2
3607	3610	RADRES
3610	8082	RADRES,
3611	8816	16/
3612	2166	POOT
3613	8631	RDISPS
3614	2241	7241/2A
3615	1577	SNAME
3616	3612	RLAB2
3617	8823	3
3620	8800	0/BLANK
3621	2240	OLABEL
3622	3631	RDISPS
3623	7241	7241/2A
3624	8764	PNAME
3625	3612	RLAB2
3626	3720	312690
3627	8764	RDISPS
3630	3617	RLAB2
3631	8803	RDISPS,
3632	8813	13/
3633	2374	PDIPPS
3634	3643	RDISPN
3635	8415	15/-
3636	2374	PDIPPS
3637	3643	RDISPN
3640	8477	77/CRR
3641	2271	PNDISP
3642	3475	STEXT

532

3643	P481	RQ1SPN1	1
3644	3128	3128/87	
3645	2372	015PN	
3646	3475	STEXT	
3647	P482	RNUM:	?
3658	P484	W/BLANK	
3651	2444	ONUM	
3652	3475	STEXT	
3653	3128	3124/78	
3654	2422	PNUM	
3655	3642	RNUH	
3656	P481	RCAN1	1
3657	4872	77/CR	
3664	2252	PCAN	
3661	3475	STEXT	
3662	4440	SIEDIT,	
3663	0844	44/0	
3664	2637	DUDU	
3665	3662	STEDIT	
3666	4451	51/1	
3667	2273	1100	
3670	3475	STEXT	
3671	P485	55/H	
3672	2662	MMDO	
3673	3662	STEDIT	
3666	4452	62/R	
3674	2662	MMDO	
3675	2662	STEXT	
3676	3475	78/X	
3677	0874	XX00	
3700	3851	STEXT	
3701	3475	43/C	
3702	6843	CCDO	
3703	3452	WILDOO	
3704	3285		
3705	4405		
3706	442		
3707	3862		
3710	3462		
3711	4472		
3712	2762		
3713	3662		
3714	4442		
3715	2761		
3716	3662		
3717	4443		
3720	3852		
3721	3662		
3722	4456		
3723	2755		
3724	3662		
3725	4454		
3726	2756		
3727	3662		
3728	4456		
3731	2755		

## One Table

SECOND TABLE  
which should begin  
with line  
WILDOO

WILDOO, 4  
SIEDIT  
47/C  
GODO  
SIEDIT  
72/E  
2200  
SIEDIT  
42/B  
R800  
SIEDIT  
43/C  
GODO  
SIEDIT  
LL00  
SIEDIT  
56/N  
KMD0  
SIEDIT  
54/L  
LL00  
SIEDIT  
DEC TAB, 2  
1513/00,  
1513

57

3643	P881	R11SPN1	1
3644	3120		3120/67
3645	2372	DISPN	
3646	3475	STEXT	
3647	6492		2
3650	P444	RNUM;	
3651	7444		0/BLANK
3652	3475	ONUM	
3653	3120	STEXT	
3654	2422		3120/70
3655	3642	PNUM	
3656	6401	RNUM	
3657	6472	RCAR:	1
3660	2252		77/GR
3661	3475	PCAR	
3662	6806	STEXT	
3663	6844	S1EDIT,	6
3664	2037		44/D
3665	3362		0000
3666	6451	STE01T	
3667	2233		51/1
3670	3475		1100
3671	6855	STEXT	
3672	2662		55/M
3673	3362	MHD0	
3674	6862	STE01T	
3675	2662		62/R
3676	3475	MHD0	
3677	6870	STEXT	
3678	2662		70/X
3701	3475	XDD0	
3702	6843	STEXT	
3703	3452	43/C	
3704	3245	CC00	
3705	6805	WILDD0,	5
3706	6842		47/G
3707	3362	GGD0	
3710	3662	STE01T	
3711	6472		72/2
3712	2762	22D0	
3713	3662	STE01T	
3714	6442		42/H
3715	2761	RB00	
3716	3662	STE01T	
3717	6443		43/C
3720	3452	CC00	
3721	3662	STE01T	
3722	6456		56/N
3723	2755	NN00	
3724	3662	STE01T	
3725	6854		54/L
3726	2756	LL00	
3727	3662	STE01T	
3730	P802	DEC1AB,	2
3731	1513		

\*400

PAL10

V142

14-JUN-74

16144 PAGE 1-32

3732 2540  
3733 3662  
3734 3120  
3735 2511  
3736 3234  
V2000  
\*2000

DECLOC  
STEDIT  
3120/70  
DECDO  
DECMB

77  
169  
7604  
3600  
7400/OFTEN USED AS -400  
320  
-200  
241  
40  
4  
4000

/MORE CONSTANTS

-12  
7400  
7400  
3770  
3770  
37  
-7  
200  
-2  
7  
17  
-240  
100  
-1  
3400  
4  
240  
260  
27  
76  
14

← INFORMATION  
OR PAGE 200  
THAT WILL  
BE MOVED  
TO PAGE 0

•400  
PAL10 V142 13-JUN-74 16:44 PAGE 1-33

(5)  
0255 7773 -5  
0256 7760 -20  
0257 3210 BIRINS  
0260 3367 ELEINS  
0261 3226 BBINIC  
0262 3442 BBBNEC  
0263 3300 EEMIC  
0264 7753 -25  
0265 3367 BBHCD  
0266 7632 BBBMC-BBBHCD

/VARIABLES

0267 8498 0  
0270 7576 7576  
0271 7575 7575  
0272 4001 4001  
0273 2494 2494  
0274 2414 2414  
0275 2494 2494  
0276 2401 2401  
0277 2494 2494  
0308 2494 2494  
0381 2494 2494  
0382 2494 2494  
0383 2494 2494  
0384 2494 2494  
0385 2494 2494  
0386 2494 2494  
0327 2494 2494  
0310 2494 2494  
0311 2494 2494  
0312 2494 2494  
0313 2494 2494  
0314 2494 2494  
0315 2494 2494  
0316 2494 2494  
0317 2494 2494  
0328 2494 2494  
0321 2494 2494  
0322 2494 2494  
0323 2494 2494  
0324 1776 1776  
0325 2494 2494  
0326 2494 2494  
0327 2494 2494  
0334 2494 2494  
0331 2494 2494  
0332 2494 2494  
0333 2494 2494  
0334 2494 2494  
0335 2494 2494  
0336 1573 1573  
0337 6681 6681

/POINTERS  
ERR0X  
BUEDIT

PAL10 V142

13-JUN-74

16144 PAGE 1-34

(56)

```

0348 1572 TOCART
0341 1123 PUPIL
0342 1121 POPKL
0343 1185 HOP
0344 3567 RINS12
0345 0544 SSUIN
0346 1261 LLESS
0347 1462 OUTRR
0354 1802 GGARBA
0356 2245 HMOVE
0357 2467 TOUT
0358 2511 MNEXT
0360 8674 STEIT
0361 3662 DUECMA
0362 2583 -LINSDO
0363 2344 NNEWIN
0364 2613 START
0365 8498 FFIND
0366 2356 *5880
0368 7281 CLA IAC
5801 3184 DCA SSL
5802 4552 JMS I OUT
5803 7364 TAD "-1
5804 1243 JMS I KNUMB
5805 4553 JMS I KNAME
5806 4555 MOTHER
5807 5421 MOTHER
5810 7402 MLT
5811 7221 CLA IAC
5812 3183 DCA SSB
5813 7281 CLA IAC
5814 3184 DCA SSN
5815 1058 TAD N24M
5816 4552 JMS I OUTB
5817 4554 JMS I POUT
5820 7482 MLT
5821 5822 MOTHER,FATHER
5822 8184 FATHEN,104
5823 3536 CDB
5824 6484
5825 6485
5826 6486
5827 6487 MASK6H, 77
5828 5829 MASK6H, 77
5830 6488 MASK6H; 168
5831 6489 N2666. 7600
5832 7600 N2666. 36000
5833 36000 N2666. 74000/OFTEN USED AS -400
5834 74000 N24MB, 74000
5835 6328 ENRUSP, 3200
5846 7600 NLG20W, 4200
5847 8241 NLG41, 241
5848 7244 NLG48, 440
5849 8884
      4

```

Test Sequence  
 for testing  
 certain parts  
 of my program

\*400

PAL10 V142

13-JUN-74

16144 PAGE 1-35

BLGCON,  
N4800, 4800  
ZU1SP,  
N4001; 488  
N26,

0015 PABA  
0016 PABA  
0017 PABA  
0020 PABA  
0021 PABA  
0022 PABA  
0023 PABA  
0024 PABA

NEXX, 0  
NLXCH, 0  
PUPMOR, 0  
PUPCON, 0  
PUPLIN, 0  
LABEL, 0  
MURU1, 0  
MURU2, 0

/MORE CONSTANTS  
NEGTEEN, -12  
NZ8000, 7000

NJ770, 3770  
N37, 37  
NEG71, -7  
N280, 280

NEG21, -1  
NEG22, -2  
N7, 7

NEG24B, -24B  
N100; 100

N1268, 3768

N3774, 3774

N8, 18  
N1609, 1609

N177, 177  
N3408, 3408

NLG1; -1  
NLG2; -2

NLG4; -4  
N24n; 24B

N260; 260

N27, 27  
NUM76, 76

NEG4, -14

NLG5; -5  
NLG2n, +2B

BLGINS, BBINS  
EELINS, EELINS

BLGMIC, BBMIC  
BLMELC, BMMEC

BLMELI, BMMELI  
ENDMIC, EEMIC

NEG25, -25  
BHICOU, BABCDO  
HICDPL, BMMIC-BBBMCD

/VARIABLES

0067 00000 CHKSUM, 0  
0070 7576 ENDWOR, 7576  
0071 7575 BEGMOR, 7575

6072 4001 ENCON1 4001  
 6073 0000 PUPY; 0  
 6074 0000 PUPX; 0  
 6075 0000 SIGN; 0  
 6076 0001 SUMIN; 1  
 6077 0000 CUDPL; 0  
 6104 0000 SSB; 0  
 6105 0000 SSNE; 0  
 6106 0000 A1; 0  
 6107 0000 B1; 0  
 6108 0000 C1; 0  
 6109 0000 K1; 0  
 6111 0000 S1; 0  
 6112 0000 V1; 0  
 6113 0000 X1; 0  
 6114 0000 W1; 0  
 6115 0000 Y1; 0  
 6116 0000 Z1; 0  
 6117 0000 2d;  
 6118 0000 MULU;  
 6120 0001 LINE1; 1  
 6121 0000 ONAME1; 0  
 6122 0000 DOL0DP; 0  
 6123 0000 GOTO; 0  
 6124 1276 LIMIT; 1776  
 6125 0000 DIRECT; 0  
 6126 0000 MICK1; 0  
 6127 0000 OPCODE; 0  
 6128 0000 WHERE; 0  
 6132 0000 NEXTY; 0  
 6133 0000 LINE1; 0  
 6134 0000 LINE2; 0  
 6135 0000 SSURE; 0  
 //POINTERS  
 6136 1573 EKOR; ERROR  
 6137 0001 BEUIT; BBEDIT  
 6140 0072 DUCART; TOCART  
 6141 123 TOPL; POPL  
 6142 121 TUPKL; POPKL  
 6143 1105 POP; KOP  
 6144 3562 RINIS2; RINST2  
 6145 0544 SURIN; SSUBIN  
 6146 1261 LLESS;  
 6147 0462 FRONTB; OUTBB  
 6150 1072 GARBAG; GGARBA  
 6151 3872 MOVE; MOVE  
 6152 0501 OUT; TOUT  
 6153 1608 WNUMH; WNUMH  
 6154 1635 PPUT; PPUT  
 6155 1216 WNAME; WNAME  
 6156 2245 NOCAR; NECCAR  
 6157 0462 OUTB; OUTBB

448

PALIN

V142

14-JUN-24

16144

PAGE 1-17

6169 6674 NEXXI MEAT  
6161 3662 SICU STUDY  
6162 2563 DECIMAL DULCIA  
6163 2384 INSDO, TINGO  
6164 6613 NEWIN, NNEWIN  
6165 9482 PITTARD STANT  
6166 2356 FINUD FFIND

\$

1440 PALIO V142 13-JUN-74 16144 PAGE 1-3A

卷之三