

.RFM \_

IDENTIFICATION

PRODUCT CODE: AC-E899D-MC  
PRODUCT NAME: CXNCADC NC-11A MODULE  
PRODUCT DATE: SEPTEMBER 1978  
MAINTAINER: DEC/X11 SUPPORT GROUP

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS MANUAL.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DIGITALS COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1975,1978 DIGITAL EQUIPMENT CORPORATION

1. ABSTRACT

"NCA" IS AN "IOMODR" THAT EXERCISES ONE NC-11A INTERFACE. THE NC-11A INTERFACE DOES MEMORY INCREMENTS VIA NPR UNTIL A WORD OR BYTE REACHES MAXIMUM CAPACITY AND ATTEMPS TO OVERFLOW. AT THIS TIME AN INTERRUPT IS GENERATED AT BR LEVEL 7. THE INTERFACE ALSO DOES TRANSFERS OF DATA TO SERIAL LOCATIONS IN CORE VIA NPR. THIS MODE IS TERMINATED BY A WORD COUNT OVERFLOW AND CONSEQUENT INTERRUPT. THE RATE OF INCREMENT OR TRANSFER IS SET BY A VARIABLE SPEED CLOCK, WHICH IS TURNED ON BY THE SWITCH ON THE BACK OF THE INTERFACE. THIS SWITCH MUST BE ON TO RUN THIS MODULE.

\* NOTE: DATA LATE ERRORS OCCUR WHEN RUNNING ON PDP-11/20 CPU \*

2. REQUIREMENTS

HARDWARE: NC-11A INTERFACE WITH MAINTENANCE SWITCH IN THE "ON" POSITION.

STORAGE:: NCA REQUIRES:  
1. DECIMAL WORDS: 2519  
2. OCTAL WORDS: 04727  
3. OCTAL BYTES: 11656

3. PASS DEFINITION

ONE PASS OF NCA MODULE CONSISTS OF FIFTY ITERATIONS OF EACH BASIC TEST SEQUENCE, WHICH RESULTS IN:

200 PROGRAM INTERRUPTS - 3,404,750 NON-PROCESSOR REQUESTS.

4. EXECUTION TIME

NCA RUNNING ALONE ON PDP-11/10 TAKES APPROXIMATELY 30 SECONDS.

5. CONFIGURATION REQUIREMENTS

DEFAULT PARAMETERS:

DEVADR: 164000, VECTOR: 270, BR1: 7, DEVCNT: 1

REQUIRED PARAMETERS:

ONLY IF PDP-11/20 CPU TYPE (REF. TO 8.)

6. DEVICE/OPTION SETUP

THE NC-11A MUST HAVE THE MAINTENANCE SWITCH IN THE "ON" POSITION.

7. MODULE OPERATION

THE FIRST MODE OF OPERATION IS WORD INCREMENT MODE.  
THIS MODE IS REPEATED FOR 50 INTERATIONS. UPON COMPLETION,  
ODD BYTE OVERFLOW MODE IS ENABLED FOR 50 INTERATIONS.  
THIS INTURN ENABLES THE EVEN BYTE MODE FOR 50 INTERATIONS.  
THIS INTURN ENABLES THE LIST MODE FOR 50 INTERATIONS.  
UPON COMPLETION, AN END OF PASS IS REPORTED AND THE MODULE IS RESTARTED.

8. OPERATION OPTIONS

SP1 IS USED TO INHIBIT TESTING MODES OF OPERATION OF THE NC11A.  
BITS 0 THRU 2 SHOULD BE SET TO A ONE IF RUNNING ON PDP-11/20 CPU.

SP1 BIT0 = 1 INHIBIT WORD INCREMENT MODE.  
SP1 BIT1 = 1 INHIBIT ODD BYTE INCREMENT MODE.  
SP1 BIT2 = 1 INHIBIT EVEN BYTE INCREMENT MODE.  
SP1 BIT3 = 1 INHIBIT LIST MODE.

9. NON STANDARD PRINTOUTS

NONE. ALL PRINTOUTS HAVE STANDARD MEANINGS AS REPRESENTED IN  
DEC/V11 DOCUMENTATION.

10. MODULE TEST ENVIROMENT

THE NCA MODULE IS KNOWN TO OPERATE UNDER THIS ENVIROMENT:

#1	POP-11/20 CPU WITH 28K	#2	PDP-11/40 CPU WITH 28K
TC11	2 DRIVES	RK11-D	1 DRIVE
TM11	1 DRIVE	TM11	1 DRIVE
TA11	2 DRIVES	NC11A	1 UNIT
NC11A	1 UNIT	AA11	1 UNIT
LP11	1 UNIT		
AA11	1 UNIT		
#3	PDP-11/10 CPU WITH 16K	#4	PDP-11/34 CPU WITH 28K
RK11-D	1 DRIVE	TC11	2 DRIVES
TA11	2 DRIVES	TM11	1 DRIVE
NC11A	1 UNIT	TA11	2 DRIVES
AA11	1 UNIT	NC11A	1 UNIT
RW11L	1 UNIT	LP11	1 UNIT
		AA11	1 UNIT
		VSV01	1 UNIT
		RK11-D	1 DRIVE

```
150 .LIST SEQ,BIN
151 IOMODR <NCAD> 164000,270,7,0,0,2,66
152 000000* MODULE 152000,NCAD,164000,270,7,0,0,2,66
153 000000* .TITLE NCAD DEC/X11 SYSTEM EXERCISER MODULE
154 ; DDXC0M VERSION 6 23-MAY-78
155 .LIST
156 *****
157 BEGIN:
158 MODNAM: .ASCII /NCAD / ;MODULE NAME
159 XFLAG: .BYTE OPEN ;USED TO KEEP TRACK OF WBUFF USAGE
160 ADDR: 164000*0 ;1ST DEVICE ADDR.
161 VECTOR: 270*0 ;1ST DEVICE VECTOR.
162 BR: .BYTE ;1ST BR LEVEL.
163 BR2: .BYTE PRTV7+0 ;2ND BR LEVEL.
164 DVID1: 0+1 ;DEVICE INDICATOR 1.
165 SR1: OPEN ;SWITCH REGISTER 1
166 SR2: OPEN ;SWITCH REGISTER 2
167 SR3: OPEN ;SWITCH REGISTER 3
168 SR4: OPEN ;SWITCH REGISTER 4
169 *****
170 STAT: 152000 ;STATUS WORD
171 INIT: STRT ;MODULE START ADDR.
172 SPOINT: MODSP ;MODULE STACK POINTER.
173 PASCNT: 0 ;PASS COUNTER
174 ICNT: 7 ;% OF ITERATIONS PER PASS=2
175 SOFCNT: 0 ;LOC TO COUNT ITERATIONS
176 HRDCNT: 0 ;LOC TO SAVE TOTAL SOFT ERRORS
177 SORPAS: 0 ;LOC TO SAVE TOTAL HARD ERRORS
178 HRDPAS: 0 ;LOC TO SAVE SOFT ERRORS PER PASS
179 SYSCNT: 0 ;LOC TO SAVE HARD ERRORS PER PASS
180 RANNUM: 0 ;% OF SYS ERRORS ACCUMULATED
181 CONFIG: 7 ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
182 RES1: C ;RESERVED FOR MONITOR USE
183 RES2: C ;RESERVED FOR MONITOR USE
184 SVR0: OPEN ;LOC TO SAVE R0.
185 SVR1: OPEN ;LOC TO SAVE R1.
186 SVR2: OPEN ;LOC TO SAVE R2.
187 SVR3: OPEN ;LOC TO SAVE R3.
188 SVR4: OPEN ;LOC TO SAVE R4.
189 SVR5: OPEN ;LOC TO SAVE R5.
190 SVR6: OPEN ;LOC TO SAVE R6.
191 CSRA: OPEN ;ADDR OF CURRENT CSR.
192 SBADR: OPEN ;ADDR OF GOOD DATA, OR
193 ACSR: OPEN ;CONTENTS OF CSR.
194 WBADR: OPEN ;ADDR OF BAD DATA, OR
195 ASSTAT: OPEN ;STATUS REG CONTENTS.
196 ERRTYP: ;TYPE OF ERROR
197 ASB: OPEN ;EXPECTED DATA.
198 AWAS: OPEN ;ACTUAL DATA.
199 RSTRT: RSTRT ;RESTART ADDRESS AFTER END OF PASS
200 WDMTO: OPEN ;WORDS TO MEMORY PER ITERATION
201 WDFRM: OPEN ;WORDS FROM MEMORY PER ITERATION
202 IWR: OPEN ;% OF INTERRUPTS PER ITERATION
203 IDNUM: 66 ;MODULE IDENTIFICATION NUMBER=66
204
```

```
205 000040 .REPT SPSIZ ;MODULE STACK STARTS HERE.
206 .NLST
207 .WORD 0
208 .LIST
209
210 000224* MODSP:
211 *****
```

```

212 000224* 000062 INTER: 50. ;SUR-SECTION INTERATION EXECUTION COUNT
213 000003 ;BUFFER ADDRESS...LOCAL TO PROGRAM
214 ;RUFVA: RUFER
215 000226* 001652* ;SPECIAL FUNCTIONS TO INTERFACE
216 ;
217 ;
218 ;
219 ;
220 000001 CLHLD=1 ;CLEAR HOLD REGISTER
221 000003 CLCOVF=2 ;CLEAR ALL OVERFLOW
222 000004 COVF=4 ;CONVERT
223 000010 CLDP=10 ;CLEAR JOY STICK DEPRESS
224 000020 CLZOVF=20 ;CLEAR Z OVERFLOW FLOP
225 000040 SW=40 ;SET TIMING MARK
226 000102 CLALL=102 ;CLEAR ALL
227 ;
228 ;
229 ;
230 ;
231 ;
232 ;
233 ;
234 ;
235 ;
236 ;
237 ;
238 ;
239 ;
240 ;
241 ;
242 ;
243 ;
244 ;
245 ;
246 ;
247 ;
248 ;
249 ;
250 ;
251 ;
252 ;
253 ;
254 ;
255 ;
256 ;
257 ;
258 ;
259 ;
260 ;
261 ;
262 ;

```

```

263 ;NOW FIND A 2048 WORD BOUNDARY ADDRESS AND SAVE IT IN LOCATION "TARGET"
264 ;
265 ;
266 ;
267 ;
268 ;
269 ;
270 ;
271 ;
272 ;
273 ;
274 ;
275 ;
276 ;
277 ;
278 ;
279 ;
280 ;
281 ;
282 ;
283 ;
284 ;
285 ;
286 ;
287 ;
288 ;
289 ;
290 ;
291 ;
292 ;
293 ;
294 ;
295 ;
296 ;
297 ;
298 ;
299 ;
300 ;

```

```

301 ;THIS IS THE PRIMING CODE.
302 ;THE INTERRUPT VECTOR IS INITIALLY SET UP TO INCREMENT WORD
303 ;TARGET UNTIL OVERFLOW AND THEN TO INTERRUPT.
304
305 PRIME: MOV #OVSR,#OVFINI ;SETUP INTERRUPT VECTOR
306 MOV #B1,#OVFINI ;LOAD BR LEVEL
307 MOV #ZOVIT1,#ZOVINT ;RESET Z OVERFLOW VECTOR
308 CLR #ZOVIT1
309 MOV #CLALL,#PASSCT ;LOAD PASS COUNT LOC.
310 MOV #CLALL,#SFUNC ;CLEAR ALL
311 BIT #BIT0,SRI ;TEST INHIBIT THIS TEST BIT
312 BNE ODDPRM ;BR IF SET TO NEXT SECTION
313 MOV #1400,#CMDCSR ;RESOLUTION SET TO 32X32X16
314 MOV #TARGET,#OFFSET ;BASE AT BUFFER TARGET LOC
315 MOV #ADDRESS,#JUNK ;TWO DUMMY READS TO
316 BIS #I01,#CMDCSR ;ENABLE OVERFLOW INTERRUPT
317 MOV #ADDRESS,#JUNK ;CLEAR BUFFER CHAIN
318 EXITS,REGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
319
320 JUNK: 0
321 PASSCT: 0
322
323 ;OVERFLOW SERVICE ROUTINES
324 ;THIS ROUTINE SERVICES 20 INTERRUPTS OF WORD OVERFLOW MODE
325
326 OVSRS: MOV #CLALL,#SFUNC ;CLEAR THE INTERFACE
327 ;-----
328 ;PIRQS,BEGIN,IS ; QUEUE UP TO CONTINUE AT IS AND RTI
329 ;-----
330 1$: MOV #CMDCSR,ACSR ;READ STATUS
331 RPL 25 ;OK
332 MOV #3,ERRTVP ;DATA LATE
333 ;*****
334 ;SOFERS,REGIN,NULL ;TIME OUT ERROR OR JOY STICK FLAG SET
335 ;*****
336 ENDS,REGIN ;
337
338 2$: MOV #-1,ASR ;LOAD "SHOULD BE" VALUE
339 MOV #TARGET,#ASADR ;LOAD "WAS ADDRESS" VALUE
340 MOV #TARGET,#AWAS ;LOAD "WAS" VALUE
341 CMP ASR,AWAS ;TEST VALUE
342 BEQ 3$ ;YES - GO ON
343 ;*****
344 ;DATERS,REGIN ;DATA ERROR!!!
345 ;*****
346 3$: DEC PASSCT ;FINISHED ?
347 BNE WORDRK ;BR IF NOT DONE

```

```

349 ;PRIME AND GO FOR TESTING ODD BYTE NPR'S
350 ;TEST LOCATION TARGET+1
351
352 ODDPRM: MOV #INTER,#PASSCT ;RESET PASS COUNT
353 MOV #ODDINT,#OVFINI ;ADJUST INT VECTOR
354 BIT #BIT1,SRI ;TEST INHIBIT THIS TEST BIT
355 BNE EVBVM ;BR IF SET TO NEXT SECTION
356 ODBVOK: CLR #TARGET ;CLEAR TARGET LOCATION
357 MOV #3400,#CMDCSR ;32X32X8 MODE
358 MOV #TARGET,#OFFSET ;SET TO BUFFER BYTE 1
359 MOV #ADDRESS,#JUNK ;LOAD OFFSET REGISTER
360 BIS #I01,#CMDCSR ;CLEAR OFF BUFFER
361 MOV #ADDRESS,#JUNK ;INTERRUPT ENABLE + GO!
362 EXITS,REGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
363
364 ODDINT: MOV #CLALL,#SFUNC ;CLEAR INTERFACE
365 ;-----
366 ;PIRQS,BEGIN,IS ; QUEUE UP TO CONTINUE AT IS AND RTI
367 ;-----
368 1$: MOV #CMDCSR,ACSP ;LOOK FOR NPR ERROR
369 RPL 25 ;OK
370 MOV #3,ERRTVP ;NPR ERROR
371 ;*****
372 ;HDRS,REGIN,NULL ;NPR-TIME OUT ERROR OR JOYSTICK FLAG
373 ;*****
374 ENDS,REGIN ;
375
376 2$: MOV #177400,ASB ;LOAD "SHOULD BE" VALUE
377 MOV #TARGET,#AWAS ;LOAD "WAS" VALUE
378 CMP ASR,AWAS ;TEST TWO VALUES
379 BEQ 3$ ;OK
380 ;*****
381 ;DATERS,REGIN ;DATA ERROR!!!
382 ;*****
383 3$: DEC PASSCT ;FINISHED ?
384 BNE ODBVOK ;BR IF NOT DONE

```

```

388 ;PRIME + GO FOR TESTING EVEN BYTE NPR'S
389 ;TEST BYTE TARGET
390
391 001164 016767 177034 177456 EVBYPW: MOV INTER,PASSCT ;RESET PASS COUNT
392 001172 012777 001256 177046 MOV #EINT,#OVFINT ;ADJUST VECTOR
393 001200 032767 000004 176610 BIT #BIT2,SRI ;TEST INHIBIT THIS TEST BIT
394 001206 001064 RNE DONE ;BR IF SET
395 001210 005077 000432 CLR #TIPGT ;LOAD WORD OVERFLOW VECTOR
396 001214 012777 003400 177006 MOV #3400,#CMDCSR ;LOAD Z OVERFLOW VECTOR
397 001222 016777 003420 177002 MOV #3400,#OVFINT ;LOAD BR LEVEL
398 001230 017767 177000 177410 CLR #TIPGT ;CLEAR INTERFACE
399 001236 052777 000161 176764 MOV #3400,#CMDCSR ;CLEAR TARGET LOCATION
400 001244 017767 176763 177374 MOV #ADDRG,JUNK ;OFFSET AT 7K
401 001252 104400 000000 000000 RIS #10,#CMDCSR ;ENABLE NC-11
402 EXITS,REGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
403 001256 012777 000102 176760 EVINT: MOV #CLALL,#SFUNC ;CLEAR INTERFACE
404 ;
405 001264 000004 000000 001272 ;
-----
406 ;IRQS,REGIN,1S ; QUEUE UP TO CONTINUE AT 1S AND RTI
-----
407 001272 017767 176732 176602 1s: MOV #CMDCSR,ACSR ;LOOK FOR NPR ERROR
408 001300 100010 BPL #3,ERRTYP ;NPR ERROR
409 001302 012767 000032 176576 MOV #32,ERRTYP ;NPR ERROR
410 ;*****
411 001310 104400 000000 000000 HDRRS,REGIN NULL ;NPR ERROR FLAG SET
412 ;*****
413 001316 104410 000000 000000 ENDS,REGIN ;
414 ;
415 001322 017767 000320 176560 2s: MOV #TARGET,AWAS ;LOAD "WAS" VALUE
416 001330 012767 000377 176550 MOV #377,ASB ;LOAD "SHOULD BE" VALUE
417 001336 026767 176544 176544 CMP ASR,AWAS ;COMPARE VALUE
418 001344 001402 BEQ #3 ;OK
419 ;*****
420 001346 104404 000000 000000 DATERS,REGIN ;DATA ERROR!!!
421 ;*****
422 ;
423 001352 005367 177272 3s: DEC PASSCT ;FINISHED ?
424 001356 001314 BNE EVBVK ;BR IF NOT DONE

```

```

425 ;PRIME FOR LIST MODE + GO
426 ;DO A TRANSFER OF 2048 WORDS AT THE CLOCK RATE THEN INTERRUPT.
427 001360 016767 176640 177262 LYSTPM: MOV INTER,PASSCT
428 001366 032767 000010 176422 BIT #BIT3,SRI ;TEST INHIBIT TEST BIT
429 001374 001117 176646 176642 RNE DONE ;BR IF SET
430 001376 001777 176646 176642 MOV #OVFINT,#OVFINT ;LOAD WORD OVERFLOW VECTOR
431 001400 005077 176640 176640 CLR #OVFINT
432 001410 012777 001502 176634 MOV #ZOVSR,#ZOVINT ;LOAD Z OVERFLOW VECTOR
433 001416 116777 176370 176630 MOV #1,#ZOVINT ;LOAD BR LEVEL
434 001420 012777 000102 176612 MOV #CLALL,#SFUNC ;CLEAR INTERFACE
435 001432 016777 000206 176574 CONTA: MOV #YMTRR,#XVHOLD ;INSERT FULL WORDS
436 001440 012777 000404 176562 MOV #404,#CMDCSR ;LIST MODE
437 001446 012777 176000 176562 MOV #2048,#WCASHI ;WORD COUNT = -2048
438 001454 016777 176546 176556 MOV #RDVY,#CAZLO ;CURRENT ADDRESS - BUFFER
439 001462 017767 176546 177156 MOV #ADDRG,JUNK ;CLEAR BUFFERS
440 001470 052777 004001 176532 RIS #4001,#CMDCSR ;ENABLE INTERRUPT AND GO
441 001476 104400 000000 000000 EXITS,REGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
442 001502 012777 000102 176534 ZOVSR: MOV #CLALL,#SFUNC ;CLEAR INTERFACE
443 ;
444 001510 000004 000000 001516 ;
-----
445 ;IRQS,REGIN,1S ; QUEUE UP TO CONTINUE AT 1S AND RTI
-----
446 001516 017767 176506 176356 1s: MOV #CMDCSR,ACSR ;LOOK FOR NPR ERROR
447 001524 100010 BPL #3,ERRTYP ;NPR ERROR
448 001526 012767 000032 176352 MOV #32,ERRTYP ;NPR ERROR
449 ;*****
450 001534 104400 000000 000000 HDRRS,REGIN NULL ;NPR ERROR FLAG SET
451 ;*****
452 001542 104410 000000 000000 ENDS,REGIN ;
453 001546 012767 001652 176330 2s: MOV #BUFFER,#ASADR ;LOAD "WAS ADDRESS"
454 001548 016767 000004 176324 MOV #YMTRR,ASB ;LOAD "SHOULD BE" DATA
455 001562 017767 176316 176320 3s: MOV #WASADR,AWAS ;GET DATA WORD
456 001570 026767 176314 176310 CMP AWAS,ASB ;CHECK FOR PROPER DATA
457 001576 001402 BEQ #4 ;BR IF CORRECT
458 ;*****
459 001600 104404 000000 000000 DATERS,REGIN ;DATA ERROR!!!
460 ;*****
461 001604 062767 000000 176272 4s: ADD #2,#WASADR ;GO TO NEXT LOCATION
462 001612 026767 176266 002052 CMP #WASADR,#BUFFER+200 ;BUT NOT TOO FAR
463 001620 001360 BNE #3 ;
464 001622 005167 000016 ;CHANCE DATA PATTERN
465 001624 005367 177016 DEC PASSCT ;PRIME WORD NPR AGAIN
466 001634 001277 CONTA
467 001634 104413 000000 000000 DONE: ENDTLS,REGIN ;SIGNAL END OF ITERATION.
468 ;MONITOR SHALL TEST END OF PASS
469 ;
470 001640 000167 176434 JMP RESTRT ;MIRROR OF X-Y HOLD
471 001644 052525 TARGET: BUFFER ;2048 WORD BOUNDARY POINTER INTO BUFFER AREA
472 001646 001652 TARGET+1 ;2048 ODD-BYTE BOUNDARY POINTER INTO BUFFER
473 001650 001653 BUFFER ;
474 001654 000000 ;
475 001654 004000 ;
476 001654 000000 ;
477 001654 000001 LAST: C ;
;END

```





REF2	000060R	184#																			
RSRT	000112R	200#																			
SADR	000122R	193#																			
SFUNC	000244R	237#	261*	292*	300*	310*	327*	365*	403*	434*	442*										
SFCNT	000042P	176#																			
SOPERS	= 104406	212#	335																		
SOPAS	000046R	178#																			
SPOI#T	000032P	172#																			
SPSIZ	= 000040	1#	205																		
SR1	000016R	165#	311	354	393	428															
SR2	000020R	179#																			
SR3	000022R	189#																			
SR4	000024R	168#																			
TART	000256R	171#	246#																		
TAT	000026R	170#																			
T#	= 000040	225#																			
SVR0	000062R	185#																			
SVR1	000064R	186#																			
SVR2	000066R	187#																			
SVR3	000070R	188#																			
SVR4	000072R	189#																			
SVR5	000074R	190#																			
SVR6	000076R	191#																			
TVSCNT	000042P	180#																			
TARGET	001646R	275#	282*	283	286	313*	315	340	341	356*	359	378	395*	397							
		415#	472#																		
ABCT1	001650R	276#	277*																		
ABDD=	000022	177#	473#																		
VECTOR	000010R	161#	255																		
MASADR	000104R	195#	340*	453*	455	461*	462														
MCSI	000236R	234#	437*																		
MDPR	000116R	202#	248*																		
MDTO	000114R	201#	247*																		
WORDBK	000600R	313#	348																		
KFLAG	00005R	158#																			
KVHLD	000234P	232#	358*	435*																	
KVMIR	001644P	435#	454	464*	471#																
ZOVINT	000222R	241#	307*	307*																	
ZOVITI	000254R	242#	256	307	308*	433*															
ZOVSR	= 001522P	432#	442#																		
	011656R	475#																			

ABS. 000000 000  
 011656 001

ERRORS DETECTED: 0  
 DEFAULT GLOBALS GENERATED: 0  
 XNCADO XNCADO/SOL/CRF:SYW=DDXCOM,XNCADO  
 RUN-TIME: 1 3 SECONDS  
 RUN-TIME RATIO: 21/3=6.0  
 CORE USED: 7K (13 PAGES)