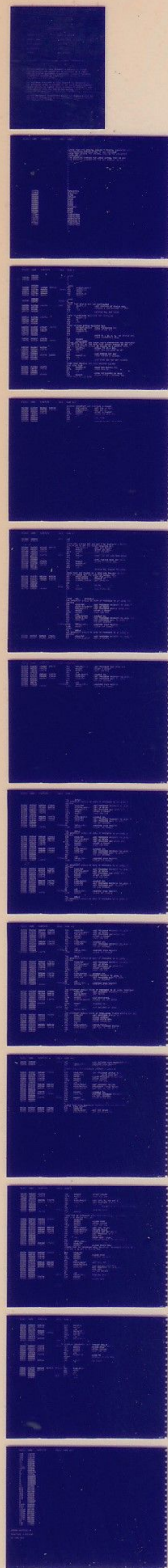


AD02

SPECIAL 1024 CHANNEL AD02
MD-11-DZADJ-A
LOGIC TEST

EP-DZADJ-A-DL
COPYRIGHT © 1976
FICHE 1 OF 1

MAY 1978
digital
MADE IN USA



IDENTIFICATION

PRODUCT CODE: MD-11-DZADJ-A-1
PRODUCT NAME: SPECIAL 1024 CH. ADØ2 LOGIC TEST
DATE CREATED: FEB. 1976
MAINTAINER: IPGCP
AUTHOR: RAY BALDWIN

Copyright: (C) 1976 DIGITAL EQUIPMENT CORPORATION.

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 DIGITAL'S 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.

```
|
|
|
|LOGIC TEST FOR SPECIAL ANALOG TO DIGITAL CONVERTER WITH
|MULTIPLEXING CAPABILITIES TO 1024 CHANNELS
|THIS TEST CHECKS THE CONTROL LOGIC OF THE CONVERTER
|HARDWARE.
|THE STARTING ADDRESS FOR LOGIC CONTROL TEST IS 200
|THE STARTING ADDRESS FOR THE DATA DISPLAY IS 210
|
```

177776
177570
000000
000240
000000
000001
000002
000003
000004
000005
000006
000007
177566
177564
176770
176772
176774
177566
177564

```
|
|PSR=177776
|SR=177570
|HLT=0
|NOP=240
|R0=X0
|R1=X1
|R2=X2
|R3=X3
|INTVC=X4
|R5=X5
|SP=X6
|PC=X7
|TYDB=177566
|TYSR=177564
|ADC8=176770
|ADDR=176772
|ADMX=176774
|TYDB=177566
|TYSR=177564
|
|
```

```

000130 000130
000130 000000
000132 000000

000200 000200
000200 012737 175520 002462
000206 012706 000600
000212 000137 000700
000216 012706 000600
000222 000137 002464

000600 000600
000600 000000
000600 000700

000700 013700 176770
000704 005700
000706 001401
000710 000000

000712 013700 176774
000716 005700
000720 001401
000722 000000

000724 012737 177677 176770
000732 013700 176770
000736 022700 100037
000742 001401
000744 000000

000746 005037 176770
000752 012737 000300 177776

000760 004367 001536
000764 105737 176770
000770 100401
000772 000000
000774 033727 176770 000001
001002 001401
001004 000000

001006 013700 176772
001012 105737 176770
001016 100001
001020 000000
001022 005037 176770

INT:      ,=130
          0
          2
          1
          1
          ,=200
LOGTST:   MOV      #=2260,ONTIME
          MOV      #STACK,X6
          JMP      ONTST
          MOV      #STACK,X6
          JMP      ONDATA

STACK:    ,=600
          0
          ,=700
TST:      ;TEST THAT STATUS REG WAS INITIALIZED
          MOV      @#ADCS,X0      ;GET CONTENTS OF STATUS REG.
          TST      X0              ;SHOULD BE INITIALIZED TO ZERO
          BEQ      ,+4
          HLT                    ;STATUS REG. NOT CLEAR.
          ;
          ;TEST MULTIPLEXER REGISTER FOR INITIALIZE
          MOV      @#ADMX,X0
          TST      X0
          BEQ      ,+4
          HLT
          ;TEST WRITING STATUS REGISTER BITS
          MOV      #177677,@#ADCS ;BILL THE STATUS REG
          MOV      @#ADCS,X0      ;READ STATUS REG
          CMP      #100037,X0
          BEQ      ,+4
          HLT                    ;BITS 0, 1, 2, 3, 4, OF STATUS REG.
          ;WERE INCORRECTLY READ.
          CLR      @#ADCS
          MOV      #300,@#PSR
          ;STALL FOR 25USEC. AND CHECK THAT A CONVERSION WAS INITIATED
          ;BY SETTING BIT 0. THE DONE FLOP (BIT07) SHOULD BE SET AND
          ;BIT 0 (A-D START) SHOULD BE CLEARED ON A CONVERT DONE.
          JSR      X3,STALL        ;STALL FOR 25 U SEC.
          TSTB    @#ADCS          ;TEST THAT A-D DONE IS SET
          BMI     ,+4
          HLT                    ;A-D DONE IS NOT SET
          BIT     @#ADCS,#1       ;TEST FOR BIT 0 CLEARED
          BEQ     ,+4
          HLT                    ;A-D START DID NOT GET CLEARED
          ;
          ;TEST THAT READING A-D DATA REGISTER CLEARS
          ;THE DONE FLOP
          MOV      @#ADDR,X0      ;READ DATA BUFFER REG.
          TSTB    @#ADCS          ;TEST DONE BIT
          BPL     ,+4
          HLT                    ;DONE NOT CLEARED BY READ
          CLR      @#ADCS         ;CLEAR STATUS & CONTROL REG
          ;TEST INIT ON A-D DONE
    
```

001026 012737 001062 000130
001034 012737 000100 176770
001042 005237 176770
001046 105737 176770
001052 100375
001054 000240
001056 000240
001060 000000

;ENABLE INTERRUPT AND INITIATE A CONVERT
MOV #RET,#INT ;SET FOR INT RET.
MOV #100,#ADCS ;SET INT EN
INC #ADCS ;START A CONVERT
TSTB #ADCS ;CHECK FOR DONE BIT
BPL .-4
NOP ;INT TIME
NOP
HLT ;FAILED TO INT, ON A DONE

```

001062 000240      RET:  NOP
001064 000240      NOP
;
;WITH INT, ENABLE SET AND BIT 7 SET INITIATE A SECOND
;CONVRTION TO SFT THE ERROR CONDITION BIT,
001066 012737 001106 000130  MOV  #RET1,#INT  ;SET INT, RETURN
001074 005237 176770      INC  #ADCS      ;START 2ND CONV.
001100 105737 176770      TSTB #ADCS     ;TEST FOR DONE
001104 100375      BPL  ,=4
001106 005737 176770      RET1: TST  #ADCS     ;TEST THAT INT WAS FROM ERROR
001112 100401      BMI  ,+4
001114 000000      HLT
001116 005037 176770      CLR  #ADCS     ;INT. WAS A=D DONE NOT ERROR
001122 013700 176772      MOV  #ADD0,#0  ;CLEAR OUT STUTS REG.
001126 005737 176770      TST  #ADCS
001132 001401      BEQ  ,+4
001134 000000      HLT
;STATUS REG, FAILED TO CLEAR,
;
;SHOW THAT INT ENABLE ON A ZERO DOES PREVENT AN INT
;SET THE ERROR BIT AND LOOK FOR ILLEGAL INT,
001136 012737 001166 000130  MOV  #RET2,#INT  ;SET FOR ILLEGAL INT
001144 012737 000100 176770  MOV  #100,#ADCS  ;ENABLE INT
001152 012737 100000 176770  MOV  #100000,#ADCS ;SET ERROR BIT
001160 000240      NOP
001162 000240      NOP
001164 000401      BR   ,+4        ;OK GO
001166 000000      HLT
001170 005037 176770      RET2: CLR  #ADCS     ;CLR STATUS REG.
;
;
;
;MACR  TAL  STATUS,N
;AN INTERRUPT SHOULD BE SEEN IF PROCESSOR IS AT LEVEL 'N'
;BIT2=0
BIS  #340,PSR      ;SET PROCESSOR PRIORITY TO LEVEL 7
MOV  #TINT'N'A,#INT ;SET UP RETURN
MOV  PSR,#INT+2    ;SET PROCESSOR STATUS AFTER INT.
INC  #ADCS        ;START A CONVERT
TSTB #ADCS        ;WAIT FOR DONE
BPL  ,=4
MOV  #100,#ADCS   ;ENABLE INT.
MOV  #STATUS,PSR  ;SET PROCESSOR PRIORITY TO LEVEL 'N'
MOV  #340,PSR     ;SET PROCESSOR BACK TO LEVEL 7
HLT  ;NO INT OCCURRED
BR   ,+4
TINT'N'A: CMP  (SP)+,(SP)+ ;RESTORE STACK POINTER
CLRB #ADCS       ;CLEAR INT, ENABLE
;ENDM
TAL  300,6
;AN INTERRUPT SHOULD BE SEEN IF PROCESSOR IS AT LEVEL 6
;BIT2=0
BIS  #340,PSR      ;SET PROCESSOR PRIORITY TO LEVEL 7
MOV  #TINT6A,#INT  ;SET UP RETURN

```

11

FI

001210	016737	176562	000132	MOV	PSR,#INT+2	;SET PROCESSOR 300 AFTER INT.
001216	005237	176770		INC	#ADCS	;START A CONVERT
001222	105737	176770		TSTB	#ADCS	;WAIT FOR DONE
001226	100375			BPL	,=4	
001230	012737	000100	176770	MOV	#100,#ADCS	;ENABLE INT.
001236	012767	000300	176532	MOV	#300,PSR	;SET PROCESSOR PRIORITY TO LEVEL 6
001244	012767	000340	176524	MOV	#340,PSR	;SET PROCESSOR BACK TO LEVEL 7
001252	000000			HLT		;NO INT OCCURRED
001254	000401			BR	,+4	
001256	022626			TINT6A: CMP	(SP)+,(SP)+	;RESTORE STACK POINTER
001260	105037	176770		CLRB	#ADCS	;CLEAR INT, ENABLE

```

TAL      240,5
;AN INTERRUPT SHOULD BE SEEN IF PROCESSOR IS AT LEVEL 5
;BIT2=0
001264 052767 000340 176504  BIS      #340,PSR      ;SET PROCESSOR PRIORITY TO LEVEL 7
001272 012737 001346 000130  MOV      #TINT5A,#INT  ;SET UP RETURN
001300 016737 176472 000132  MOV      PSR,#INT+2    ;SET PROCESSOR 240 AFTER INT.
001306 005237 176770          INC      #ADCS         ;START A CONVERT
001312 105737 176770          TSTB    #ADCS         ;WAIT FOR DONE
001316 100375          BPL      #-4
001320 012737 000100 176770  MOV      #100,#ADCS    ;ENABLE INT.
001326 012767 000240 176442  MOV      #240,PSR     ;SET PROCESSOR PRIORITY TO LEVEL 5
001334 012767 000340 176434  MOV      #340,PSR     ;SET PROCESSOR BACK TO LEVEL 7
001342 000000          HLT
001344 000401          BR      #+4
001346 022626          TINT5A: CMP      (SP)+,(SP)+ ;RESTORE STACK POINTER
001350 105037 176770          CLRB    #ADCS        ;CLEAR INT. ENABLE
    
```

```

TAL      200,4
;AN INTERRUPT SHOULD BE SEEN IF PROCESSOR IS AT LEVEL 4
;BIT2=0
001354 052767 000340 176414  BIS      #340,PSR      ;SET PROCESSOR PRIORITY TO LEVEL 7
001362 012737 001436 000130  MOV      #TINT4A,#INT  ;SET UP RETURN
001370 016737 176402 000132  MOV      PSR,#INT+2    ;SET PROCESSOR 200 AFTER INT.
001376 005237 176770          INC      #ADCS         ;START A CONVERT
001402 105737 176770          TSTB    #ADCS         ;WAIT FOR DONE
001406 100375          BPL      #-4
001410 012737 000100 176770  MOV      #100,#ADCS    ;ENABLE INT.
001416 012767 000200 176352  MOV      #200,PSR     ;SET PROCESSOR PRIORITY TO LEVEL 4
001424 012767 000340 176344  MOV      #340,PSR     ;SET PROCESSOR BACK TO LEVEL 7
001432 000000          HLT
001434 000401          BR      #+4
001436 022626          TINT4A: CMP      (SP)+,(SP)+ ;RESTORE STACK POINTER
001440 105037 176770          CLRB    #ADCS        ;CLEAR INT. ENABLE
    
```

```

TAL      140,3
;AN INTERRUPT SHOULD BE SEEN IF PROCESSOR IS AT LEVEL 3
;BIT2=0
001444 052767 000340 176324  BIS      #340,PSR      ;SET PROCESSOR PRIORITY TO LEVEL 7
001452 012737 001526 000130  MOV      #TINT3A,#INT  ;SET UP RETURN
001460 016737 176312 000132  MOV      PSR,#INT+2    ;SET PROCESSOR 140 AFTER INT.
001466 005237 176770          INC      #ADCS         ;START A CONVERT
001472 105737 176770          TSTB    #ADCS         ;WAIT FOR DONE
001476 100375          BPL      #-4
001500 012737 000100 176770  MOV      #100,#ADCS    ;ENABLE INT.
001506 012767 000140 176262  MOV      #140,PSR     ;SET PROCESSOR PRIORITY TO LEVEL 3
001514 012767 000340 176254  MOV      #340,PSR     ;SET PROCESSOR BACK TO LEVEL 7
001522 000000          HLT
001524 000401          BR      #+4
001526 022626          TINT3A: CMP      (SP)+,(SP)+ ;RESTORE STACK POINTER
001530 105037 176770          CLRB    #ADCS        ;CLEAR INT. ENABLE
    
```

```

TAL      100,2
;AN INTERRUPT SHOULD BE SEEN IF PROCESSOR IS AT LEVEL 2
;BIT2=0
    
```



```

001534 052767 000340 176234      BIS      #340,PSR      ;SET PROCESSOR PRIORITY TO LEVEL 7
001542 012737 001616 000130      MOV      #TINT2A,#INT  ;SET UP RETURN
001550 016737 176222 000132      MOV      PSR,#INT+2  ;SET PROCESSOR 100 AFTER INT.
001556 005237 176770      INC      #ADCS      ;START A CONVERT
001562 105737 176770      TSTB    #ADCS      ;WAIT FOR DONE
001566 100375      BPL      ,-4
001570 012737 000100 176770      MOV      #100,#ADCS  ;ENABLE INT.
001576 012767 000100 176172      MOV      #100,PSR   ;SET PROCESSOR PRIORITY TO LEVEL 2
001604 012767 000340 176164      MOV      #340,PSR   ;SET PROCESSOR BACK TO LEVEL 7
001612 000000      HLT
001614 000401      BR      ,+4
001616 022626      TINT2A: CMP    (SP)+,(SP)+ ;RESTORE STACK POINTER
001620 105037 176770      CLRB    #ADCS      ;CLEAR INT. ENABLE

TAL      40,1
;AN INTERRUPT SHOULD BE SEEN IF PROCESSOR IS AT LEVEL 1
;BIT2=0
001624 052767 000340 176144      BIS      #340,PSR      ;SET PROCESSOR PRIORITY TO LEVEL 7
001632 012737 001706 000130      MOV      #TINT1A,#INT  ;SET UP RETURN
001640 016737 176132 000132      MOV      PSR,#INT+2  ;SET PROCESSOR 40 AFTER INT.
001646 005237 176770      INC      #ADCS      ;START A CONVERT
001652 105737 176770      TSTB    #ADCS      ;WAIT FOR DONE
001656 100375      BPL      ,-4
001660 012737 000100 176770      MOV      #100,#ADCS  ;ENABLE INT.
001666 012767 000040 176102      MOV      #40,PSR    ;SET PROCESSOR PRIORITY TO LEVEL 1
001674 012767 000340 176074      MOV      #340,PSR   ;SET PROCESSOR BACK TO LEVEL 7
001702 000000      HLT
001704 000401      BR      ,+4
001706 022626      TINT1A: CMP    (SP)+,(SP)+ ;RESTORE STACK POINTER
001710 105037 176770      CLRB    #ADCS      ;CLEAR INT. ENABLE

;
;NO INTERRUPT SHOULD OCCUR IF PROCESSOR IS AT LEVEL 7(BIT2=0)
001714 052767 000340 176054      BIS      #340,PSR      ;SET PROCESSOR PRIORITY TO LEVEL 7
001722 012737 001706 000130      MOV      #RET3,#INT  ;SET RETURN
001730 016737 176042 000132      MOV      PSR,#INT+2
001736 105037 176770      CLRB    #ADCS      ;CLR STATUS REG.
001742 012737 100100 176770      MOV      #100100,#ADCS ;ENABLE INT. SET ERROR
001750 000240      NOP
001752 105037 176770      CLRB    #ADCS      ;DISABLE INT.
001756 000402      BR      ,+6
001760 000000      RET3:  HLT
001762 022626      CMP     (%)+,(%)+

;
;NO INTERRUPT SHOULD OCCUR AT PROC. LEVEL 7 WITH STATUS BIT 2=1
001764 052767 000340 176004      BIS      #340,PSR      ;SET PROCESSOR STATUS TO 7
001772 005737 176772      TST     #ADDR      ;CLEAR DONE BIT
001776 005037 176770      CLR     #ADCS      ;CLEAR STATUS REG.
002002 005237 176770      INC     #ADCS      ;START CONVERSION
002006 100375      BPL      ,-4
002010 012737 002036 000130      MOV      #RET4,#INT  ;SETUP RETURN
002016 012737 000340 000132      MOV      #340,#INT+2 ;RETURN PROCESSOR STATUS
002024 012737 000104 176770      MOV      #104,#ADCS  ;SET INT. EN. AND BIT 2
002032 000240      NOP
002034 000402      BR      ,+6

```

002036	000000			RET4:	HLT		;INT OCCURRED WITH PROCESSOR AT 7
002040	022626				CMP	(SP)+,(SP)+	;RESTORE STACK POINTER
002042	005037	176770			CLR	0#ADCS	;CLEAR STATUS REG.
						;CHECK INT, WITH PROCESSOR PRIORITY AT LEVEL 0	
002046	052767	000340	175722		BIS	#340,PSR	;SET PROCESSOR STATUS TO 7
002054	013700	176772			MOV	0#ADDR,X0	;CLEAR DONE BIT
002060	005037	177776			CLR	0#PSR	;CLEAR PROCESSOR STATUS REG.
002064	005237	176770			INC	0#ADCS	;START CONVERT
002070	105737	176770			TSTB	0#ADCS	;WAIT FOR DONE
002074	100375				BPL	,=4	
002076	012737	002136	000130		MOV	#RETS,0#INT	;SET INTERRUPT RETURN
002104	012737	000340	000132		MOV	#340,0#INT+2	;PROCESSOR STATUS RETURN
002112	012737	000104	176770		MOV	#104,0#ADCS	;SET INT, EN, AND BIT 2
002120	000240				NOP		
002122	005037	176770			CLR	0#ADCS	;DISABLE INT.
002126	052767	000340	175642		BIS	#340,PSR	;RESTORE PROCESSOR PRIORITY
002134	000000				HLT		;NO INT, OCCURRED
002136	005037	176770		RETS:	CLR	0#ADCS	;CLEAR STATUS REG.
002142	022626				CMP	(SP)+,(SP)+	
002144	000240				NOP		
						;CHECK THAT WITH PROCESSOR PRIORITY AT 5 AND BIT 2 SET	
						;AN INT WILL OCCUR.	
002146	052767	000340	175622		BIS	#340,PSR	
002154	012737	002224	000130		MOV	#RETS,0#INT	;SET INT RETURN
002162	012737	000340	000132		MOV	#340,0#INT+2	;SET RETURN STATUS

```

002170 005237 176770      INC      @NADCS      ;START CONVERT
002174 105737 176770      TSTB    @NADCS      ;WAIT FOR DONE
002200 100375                BPL      ,=4
002202 012737 000104 176770  MOV      #104,@NADCS  ;SET INT, EN, AND BIT 2
002210 012767 000240 175560  MOV      #240,PSR    ;SET PROCESSOR STATUS TO 5
002216 000240                NOP
002220 000402                BR       ,+6        ;FAILED TO INT A LEVEL 5 BIT2=1
002222 000240                NOP
002224 000000      RET6:  HLT
002226 005037 176770      CLR      @NADCS      ;RESTORE STACK
;
;TEST FOR NO INTERRUPT WITH PROCESSOR STATUS AT 6 AND
;BIT 2 OF STATUS REG, SET
002232 052767 000340 175536  BIS      #340,PSR
002240 005737 176772      TST     @NADDR      ;CLEAR DONE
002244 005237 176770      INC     @NADCS      ;START CONVERT
002250 105737 176770      TSTB   @NADCS      ;WAIT FOR DONE
002254 100375                BPL      ,=4
002256 012737 002412 000130  MOV      #RET7,@INT  ;SET INT, RETURN
002264 012737 000340 000132  MOV      #340,@INT+2 ;RETURN STATUS
002272 012737 000104 176770  MOV      #104,@NADCS ;ENABLE INT, SET BIT 2
002300 012767 000300 175470  MOV      #300,PSR    ;SET PROCESSOR AT 6
002306 000240                NOP
002310 000402                BR       ,+6        ;SKIP NEXT 2 INST.
002312 000000      HLT
002314 022626                CMP      (SP)+,(SP)+ ;SHOULD NOT HAVE INT AT LEVEL 6
002316 012767 000340 175452  MOV      #340,PSR
002324 005037 176770      CLR     @NADCS      ;CLEAR STATUS REG.
;CHECK THAT AN INTERRUPT WILL OCCUR WHEN PROCESSOR STATUS IS 4
;AND BIT 2 IS SET IN STATUS REG.
;
002330 052767 000340 175440  BIS      #340,PSR
002336 005737 176772      TST     @NADDR      ;CLEAR DONE
002342 005237 176770      INC     @NADCS      ;CONVERT AND WAIT.
002346 105737 176770      TSTB   @NADCS
002352 100375                BPL      ,=4
002354 012737 002412 000130  MOV      #RET7,@INT  ;SET INT, RET.
002362 012737 000340 000132  MOV      #340,@INT+2
002370 012737 000104 176770  MOV      #104,@NADCS ;SET INT EN, AND BIT 2
002376 012767 000200 175372  MOV      #200,PSR    ;SET PROCESSOR TO LEVEL 4
002404 000240                NOP
002406 000000      HLT
002410 000401                BR       ,+4
002412 022626                CMP      (SP)+,(SP)+ ;RESTORE STACK
002414 005037 176770      CLR     @NADCS
002420 005737 176772      TST     @NADDR      ;CLEAR DONE

```

```

002424 005037 000132      CLR      @INT+2
002430 005337 002462      DEC      @TIME
002434 001402              BEQ      ,+6
002436 000137 000206      JMP      @LOGTST
002442 012737 000007 177566    MOV      @7,@TYDB
002450 105737 177564      TSTB    @TYSR
002454 100375              BPL      ,=4
002456 000137 000200      JMP      @LOGTST-6
002462 000000              TIME: 0
;
; DISPLAY CONVERTED DATA THROUGH REG. 0
002464 005037 176770      DATA: CLR      @ADCS      ;CLEAR STATUS REG
002470 005737 176772      TST      @ADDR      ;CLEAR DONE
002474 005237 176770      INC      @ADCS      ;START CONVERT
002500 105737 176770      TSTB    @ADCS      ;WAIT FOR DONE
002504 100375              BPL      ,=4
002506 013700 176772      MOV      @ADDR,@0    ;LOAD R0
002512 000005              RESET
002514 000240              NOP
002516 000137 002464      JMP      @DATA      ;DO AGAIN
;
;
002522 012737 000140 002540  STALL: MOV      @140,@DEC
002530 005337 002540      DEC      @DEC
002534 001375              BNE      ,=4
002536 000203              RTS      %3
002540 000000              DEC: 0
;
; .END

```

ADCS 176770
ADDR 176772
ADMX 176774
DATA 002464
DEC 002540
HLT 000000
INT 000130
INTVC 000004R
LOGTST 000206
NOP 000240
PC 000007R
PSR 177776
R0 000000R
R1 000001R
R2 000002R
R3 000003R
R5 000005R
RET 001062
RET1 001106
RET2 001166
RET3 001760
RET4 002036
RET5 002136
RET6 002224
RET7 002412
SP 000006R
SR 177570
STACK 000600
STALL 002522
TIME 002462
TINT1A 001706
TINT2A 001616
TINT3A 001526
TINT4A 001436
TINT5A 001346
TINT6A 001256
TST 000700
TYDB 177566
TYSR 177564

ERRORS DETECTED: 0

RUN-TIME: 2 SECONDS

5K CORE USED