

KXT11-CA ROM Listing

Prepared by Educational Services
of
Digital Equipment Corporation

First Edition, June 1984

© Digital Equipment Corporation 1984.

All Rights Reserved.

Printed in U.S.A.

This software is furnished under a license and may be used and copied only in accordance with the terms of such license and with the inclusion of the above copyright notice. This software or any other copies thereof may not be provided or otherwise made available to any other person. No title to and ownership of the software is hereby transferred.

The information in this software is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation.

Digital assumes no responsibility for the use or reliability of its software on equipment that is not supplied by Digital.

The following are trademarks of Digital Equipment Corporation:

digital	MASSBUS	TOPS-10
DEC	PDP	TOPS-20
DECmate	P/OS	UNIBUS
DECsystem-10	Professional	VAX
DECSYSTEM-20	Q-Bus	VMS
DECUS	Rainbow	VT
DECwriter	RSTS	Work Processor
DIBOL	RSX	

CONTENTS

INTRODUCTION.....	1
KXTDEF.MAC - Dual-Port RAM Definitions on KXT11-CA.....	5
NFDEF - Native Firmware Definitions.....	10
SELF TEST OVERLAY 0.....	13
- T1 I/O REGISTER CHECK.....	19
- T2 NATIVE (AND USER) RAM.....	21
- T3 NATIVE (AND USER) ROM.....	24
- T4 T11 CPU INSTRUCTIONS AND TRAPS.....	26
- T5 LINE CLOCK (BEVNT) INTERRUPT.....	27
- T6 CONSOLE SERIAL PORT DC319.....	29
- T7 SECOND SERIAL PORT NEC7201.....	31
LBTEST - LOOP BACK TESTS.....	41
RESTRRT - RESTART HANDLER.....	43
SAVERG - SAVE REGISTERS.....	47
SODTM - SERIAL ODT MONITOR.....	51
ADDREG - GET ADDRESS/REGISTER.....	59
EXAMIN - EXAMINE.....	63
DISPST - DISPLAY STRING.....	67
DATCOM - GET DATA/COMMAND.....	70
DEPOS - DEPOSIT.....	72
DISPWD - DISPLAY WORD.....	74
ABAUD - AUTO BAUD.....	77
GOPROC - GO/PROCEED.....	80
SETLED - SET LEDS.....	81
TRAP4 - TRAP 4 EMULATOR.....	84
TRAP10 - TRAP 10 EMULATOR.....	85
TRAP24 - TRAP 24 EMULATOR.....	86
TRAPX - TRAP COMMAND EMULATOR.....	87
WAITST - WAIT FOR COMMAND STATE.....	88
QCOMIN - Q-BUS COMMAND INTERP.....	90
DECTST - DECODE AND DISPATCH Q-BUS SELF TEST REQUESTS.....	95
TNNERR - UPDATE LEDS AFTER SELFTEST NN.....	99
DMAULD - DMA LOADER.....	102
BSUPV - BOOT SUPERVISOR.....	104
QODTM - Q-BUS ODT MONITOR.....	107
TU58BT - TU58 BOOT STRAP.....	110
PUTEST - POWER-UP SELF TEST.....	114
EXTEST - EXECUTE TEST.....	116
ENTRY POINT.....	119
PWRUP - POWER-UP HANDLER.....	120
INIT - INITIALIZATION.....	122
DINIT - DEVICE INITIALIZATION.....	124
ACCUM - ACCUMULATE NUMBER.....	125
GETIN - GET INPUT.....	126
CHKCHR - CHECK CHARACTER.....	127
SPR5SU - SP/R5 SETUP.....	129
VERS - VERSION NUMBER AND TEST.....	130
SELF TEST OVERLAY 1.....	131
- T10 PARALLEL I/O PORT Z8036.....	135
- T11 DMA CONTROLLER AMZ8016.....	140
- T12 Q-BUS INTERRUPT.....	145
- T13 DUAL-PORT RAM AND IPV.....	147

INTRODUCTION

This is a general overview of the KXTNT KXT11-CA-macro V05.00 native firmware implementation.

The firmware was written in several modules which were assembled separately and linked to produce the complete native firmware. For purposes of generating a listing for publication, a command procedure was developed which converts these modules into a single module which, when assembled, yields a listing that contains the actual addresses in the address column and the operand field.

The following is a list of each of the modules and a brief description of the function provided by each. A more detailed description proceeds each module:

AUD (Auto Baud) - Performs the auto baud function for serial ODT

ACCUM (Accumulate Number) - Accumulates a 16-bit number from octal input

ADDREG (Get Address/Register Number) - Inputs an address or register number for serial ODT

BSUPV (Boot Supervisor) - Controls the boot process

CHKCHR (Check Character) - Polls for input from console line

DATCOM (Data/Command) - Inputs data and commands for serial ODT

DECTST (Decode Test) - Decodes Q-Bus controlled test commands

DEPOS (Deposit) - Performs the deposit function for serial and Q-Bus ODTs

DINIT (Device Initialization) - Initializes all local I/O devices

DISPST (Display String) - Sends a print string to the console serial port

DISPWD (Display Word) - Converts a 16-bit word to an ASCII string and sends the string out to the console serial port

DMALD (DMA Load) - Controls the execution of the DMA load command

ENTRY (Entry Points) - Defines the two entry points to the native firmware (173000, 173004)

EXAMIN (Examine) - Performs the ODT examine function

EXTEST (Execute Test) - Executes each of the self tests

GETIN (Get Input) - Waits for and gets input from console serial line

GOPROC (Go/Proceed) - Performs both the Go and Proceed function of ODT

INIT (Initialization) - Initializes the local I/O

KXTDEF (KXT Definitions) - Defines the KXT specific constants used by the native firmware, KUI and MicroPower

LBTEST (Loop Back Test) - Performs all the self tests executed when in the loop back test mode

NFDEF (Native Firmware Definitions) - Defines all general native firmware related constants

PUTEST (Power-Up Tests) - Performs the appropriate power-up self tests

— PWRUP (Power-Up Interrupt Handler) - Top level routine entered by Power-Up interrupt (173000)

QCOMIN (Q-Bus Command Interpreter) - Decode Q-Bus commands passed to first word of two-port RAM

QODTM (Q-Bus ODT Monitor) - Controls Q-Bus ODT

RESTRRT (Restart Interrupt Handler) - Top level routine entered by Restart interrupt (173004)

SAVERG (Save Registers) - Save the user context of the CPU registers on entry of ODT

SETLED (Set LEDs) - Controls the LED display

SLFTSO (Self Test Overlay 0) - Normally mapped section of self tests (tests 1 through 7)

— SLFTS1 (Self Test Overlay 1) - Overlay section of self tests (tests 10 through 13)

SODTM (Serial ODT Monitor) - Controls serial ODT

SPR5SU (Stack Pointer/R5 Setup) - initializes stack pointer and R5 for particular memory map selection

TNNERR (Test N Error Decoder) - Decodes error from given test

TRAP10 (Trap 10 Emulator) - Emulates trap to 10

TRAP24 (Trap 24 Emulator) - Emulates trap to 24

TRAP4 (Trap 4 Emulator) - Emulates trap to 4

TRAPX (Trap X Emulator) - Emulates trap to address passed in two-port RAM

TU58BT (TU58 Boot) - TU58 primary loader

VERS (Version) - Version code definition

WAITST (Wait State) - Wait for Q-Bus command loop

FIRMWARE MEMORY ORGANIZATION

The native firmware is contained in a 4 K word set of ROMs which are mapped in a 3 K word space with 1 K of overlay. It is organized into five sections in the native ROM. Table 1 describes these sections.

Table 1 Memory Organization

KXT Memory Address	ROM Address	Description
160000 - 163777	0000 - 3777	Normally mapped section of overlay memory (OVL0)
164000 - 172777	4000 - 12777	Permanently mapped section 1 (PERM1)
173000 - 173007	14000 - 14007	Interrupt entries (VECT)
173010 - 173777	14010 - 14777	Permanently mapped section 2 (PERM2)
160000 - 163777	15000 - 17777	Overlay section 1 (OVL1)

IMPORTANT ADDRESSES

The addresses listed in Table 2 are of special interest. Any address specified by symbolic name can be determined by looking that name up in the symbol list at the end of the native firmware listing.

Table 2 Special Addresses

Name	Module	Description
PERV(173000)	ENTRY	Power-up interrupt address
RSTV(173004)	ENTRY	Restart interrupt address
FARSTR	PUTEST	Fatal self test error loop
FAS	BSUPV	Fatal error loop when invalid Boot/Self Test switch position set
FAR	RESTART	Loop address after Restart interrupt when a fatal error condition exists

REGISTER USE

The only CPU register that has a fixed use throughout the native firmware is R5. This register is used to point to the top of the native RAM (the highest word address) and is referenced throughout the firmware. This register should never be altered while the firmware is executing.

```
5 .SBTTL KXTDEF.MAC - Dual Port RAM definitions on KXT11CA
6 .ident /V1.5/
7 ;
8 ; FUNCTIONAL DESCRIPTION:
9 ;
10 ; This module defines addresses and bit definitions for various registers
11 ; on the KXT11-CA. The following naming conventions are used:
12 ;
13 ; KW.xxx = Word addressable register on the KXT11-C
14 ; KL.xxx = Low byte addressable register on the KXT11-C
15 ; KH.xxx = High byte addressable register on the KXT11-C
16 ; KXSxxx = Read/write bit definition for a KXT11-C register
17 ; KWSxxx = Write only bit definition for a KXT11-C register
18 ; KRSxxx = Read only bit definition for the KXT11-C register
19 ; KHSxxx = Multi-bit mask for a KXT11-C register
20 ; KPSxxx = Bit Pattern for Multi-bit fields of KXT11-C registers
21 ;
22 ; These definitions can be used in one of two ways. They can be accessed
23 ; by using the macro KXTDF$ which will define each symbol locally in the
24 ; module using the macro. Secondly, this module can be assembled which will
25 ; define each symbol globally in an object that other modules can link to.
26 ; A KXT kernel includes the global definitions so any application merging
27 ; with a KXT kernel need not use the local macro.
28 ;
29 ;-
30 ;
31 ; IF THIS MODULE IS ASSEMBLED, THEN GENERATE THE SYMBOLS IN A GLOBAL MODE
32 ;> and show the macro expansions
33 .list me
34 KXTD$ <==:>
35 ;
36 ; KXT11-C CSR A definitions
37
38 177520 KW.CSA ==: 177520 ; KXTCSRA word location
39 177520 KL.CSA ==: 177520 ; KXTCSRA low byte location
40
41 000001 KX$ODT ==: 1 ; Sync Mode on channel B enable
42 000002 KX$2EN ==: 2 ; SLU 2 read enable
43 000004 KX$SMA ==: 4 ; Sync Mode on channel A enable
44 000010 KX$TTC ==: 10 ; Terminal connect
45 000020 KX$TIS ==: 20 ; Terminal in service
46 000040 KX$DPE ==: 40 ; Diagnostic PROM enable
47 000100 KX$RTE ==: 100 ; Real-time clock interrupt enable
48 000200 KX$CTE ==: 200 ; Counter interrupt enable
49
50 ; KXT11 CSR B definitions
51
52 177522 KW.CSB ==: 177522 ; Word address for KXTCSR B
53 177522 KL.CSB ==: 177522 ; Low byte address for KXTCSR B
54
55 000001 KRSTNS ==: 1 ; Terminal in normal service mode
56 000016 KM$MAP ==: 16 ; Memory map configuration
57 000002 KX$MPO ==: 2 ; Memory map bit 0
58 000004 KX$MP1 ==: 4 ; " " " 1
59 000010 KX$MP2 ==: 10 ; " " " 2
60 000360 KM$SWS ==: 360 ; Boot/Self Test switch position
```

; KXT11 CSR C definitions

177524	KW.CSC ==:	177524	; Word address for KXTCSR C
177524	KL.CSC ==:	177524	; Low byte address for KXTCSR C
000001	KX\$LD0 ==:	1	; LED 0
000002	KX\$LD1 ==:	2	; LED 1
000004	KX\$LD2 ==:	4	; LED 2
000010	KX\$LD3 ==:	10	; LED 3
000360	KMSIDS ==:	360	; KXT11-C ID switch setting

; KXT11 CSR D definitions

177530	KW.CSD ==:	177530	; Word address for KXTCSR D
177530	KL.CSD ==:	177530	; Low byte address for KXTCSR D
177531	KH.CSD ==:	177531	; High byte address for KXTCSR D
000001	KX\$RD0 ==:	1	; Dualport RAM interrupt 120 request
000002	KX\$RD4 ==:	2	; Dualport RAM interrupt 124 request
000004	KX\$RD8 ==:	4	; Dualport RAM interrupt 134 request
000010	KX\$R10 ==:	10	; Dualport RAM interrupt 120 enable
000020	KX\$R14 ==:	20	; Dualport RAM interrupt 124 enable
000040	KX\$R18 ==:	40	; Dualport RAM interrupt 134 enable
000100	KX\$DEN ==:	100	; Dualport RAM enable
000200	KX\$DRT ==:	200	; Dualport RAM non-maskable trap
000400	KX\$BHE ==:	400	; B-halt enable
001000	KX\$BHF ==:	1000	; b-halt flag
002000	KX\$ORE ==:	2000	; Q-bus reset enable
004000	KX\$QRF ==:	4000	; Q-bus reset
010000	KX\$BQI ==:	10000	; Block Q bus interrupt register
020000	KX\$QIE ==:	20000	; QIR interrupt enable
040000	KRSQRP ==:	40000	; QIR request pending
100000	KX\$NXM ==:	100000	; Non-existent memory flag

; KXT11 I/O Buffer Control definitions

177540	KW.IOC ==:	177540	; Word address for IOBFR C
177540	KL.IOC ==:	177540	; Low byte address for IOBFR C
177541	KR.IOC ==:	177541	; High byte address for IOBFR C
000001	KW\$PB0 ==:	1	; PIO port B byte 0 direction
000002	KW\$PB1 ==:	2	; PIO port B byte 1 direction
000004	KW\$PB2 ==:	4	; PIO port B byte 2 direction
000010	KW\$PB3 ==:	10	; PIO port B byte 3 direction
000020	KW\$PB4 ==:	20	; PIO port B byte 4 direction
000040	KW\$PB5 ==:	40	; PIO port B byte 5 direction
000100	KW\$PB6 ==:	100	; PIO port B byte 6 direction
000200	KW\$PB7 ==:	200	; PIO port B byte 7 direction
000400	KW\$PLD ==:	400	; PIO port A low-byte direction
001000	KW\$PBD ==:	1000	; PIO port A high-byte direction
002000	KW\$POD ==:	2000	; PIO port C word 0 direction
004000	KW\$PIN ==:	4000	; PIO port C word 1 direction
010000	KW\$P2D ==:	10000	; PIO port C word 2 direction
020000	KW\$P3D ==:	20000	; PIO port C word 3 direction
040000	KW\$PAB ==:	40000	; PIO ports A and B active pull-up

100000	KW\$PC0 ==:	100000	; PIO port C active pull-up
; DUAL PORT RAM IOP COMMAND REGISTER DEFINITIONS			
175000	KW.CMD ==:	175000	;KXT DPR REGISTER 0 WORD LOC.
175000	KL.CMD ==:	175000	; " " " LOW BYTE LOC.
175001	KH.CMD ==:	175001	; " " " HIGH BYTE LOC.
; SYSTEM COMMAND BIT DEFINITIONS			
000001	KX\$TRC ==:	1	;TRAP COMMAND
000002	KX\$DMC ==:	2	;DMA LOAD COMMAND
000004	KX\$INC ==:	4	;RE-INITIALIZE COMMAND
000010	KX\$ODC ==:	10	;ENTER Q BUS ODT MODE COMMAND
000020	KX\$SHC ==:	20	;SHOW CONFIGURATION COMMAND
000040	KX\$NOP ==:	40	;NO OPERATION COMMAND
100001	KX\$T01 ==:	100001	;CSP TEST COMMAND
100002	KX\$T02 ==:	100002	;RAM " "
100004	KX\$T03 ==:	100004	;ROM " "
100010	KX\$T04 ==:	100010	;CPU " "
100020	KX\$T05 ==:	100020	;BEVENT " , "
100040	KX\$T06 ==:	100040	;SLU1 " "
100100	KX\$T07 ==:	100100	;SLU2 " "
100200	KX\$T10 ==:	100200	;PARALLEL I/O TEST COMMAND
100400	KX\$T11 ==:	100400	;DMA TEST COMMAND
101000	KX\$T12 ==:	101000	;QIR " "
102000	KX\$T13 ==:	102000	;DPR " "
; Q BUS ODT COMMAND DEFINITIONS			
000001	KX\$OMO ==:	1	;OPEN AND EXAMINE MEMORY COMMAND
000002	KX\$ORO ==:	2	;OPEN AND EXAMINE REGISTER COMMAND
000004	KX\$DEO ==:	4	;DEPOSIT COMMAND
000010	KX\$COO ==:	10	;GO COMMAND
000020	KX\$PRO ==:	20	;PROCEED COMMAND
100000	KX\$EXO ==:	100000	;EXIT ODT COMMAND
; DUAL PORT RAM IOP STATUS REGISTER DEFINITIONS			
175002	KW.STA ==:	175002	;KXT DPR REGISTER 1 WORD LOC.
175002	KL.STA ==:	175002	; " " " LOW BYTE LOC.
175003	KH.STA ==:	175003	; " " " HIGH BYTE LOC.
000007	KM\$STF ==:	7	;STATE FIELD
000000	KP\$INI ==:	0	;INITIALIZATION STATE
000001	KP\$STS ==:	1	;POWER UP SELF TEST STATE
000002	KP\$OTS ==:	2	;Q BUS CONTROLLED TEST MODE
000003	KP\$QOD ==:	3	;Q BUS ODT MODE
000004	KP\$WST ==:	4	;WAITING FOR COMMAND STATE
000005	KP\$PBS ==:	5	;PRIMARY BOOTSTRAP STATE
000007	KP\$NNC ==:	7	;EXECUTING NON NATIVE CODE
000010	KX\$SEF ==:	10	;STACK ERROR FLAG

KXTNF - KXT11CA Native Firmware MACRO V05.00 00:51 Page 3-3
 KXTDEF.MAC - Dual Port RAM definitions on KXT11CA

```

000020          KX$OHL ==: 20           ;ENTER ODT ON HALT FLAG
000040          KX$FEF ==: 40           ;FATAL ERROR FLAG
000100          KX$SOF ==: 100          ;SERIAL ODT FLAG
000200          KX$QOF ==: 200          ;Q BUS ODT FLAG
001000          KX$PWF ==: 1000         ;POWER UP NO BATTERY BACKUP FLAG
002000          KX$BAT ==: 2000         ;BATTERY BACKUP POWER UP FLAG
004000          KX$SXT ==: 4000         ;STACK POINTER NXM TEST FLAG
010000          KX$NKF ==: 10000        ;NXM HANDLING FLAG
020000          KX$BDF ==: 20000        ;BREAK DISABLE FLAG
040000          KX$DME ==: 40000        ;DMA LOAD ERROR FLAG
100000          KX$CME ==: 100000       ;COMMAND ERROR FLAG

;
; DUAL PORT RAM SYSTEM CONTROL REGISTER 2
;

175004          KW.SC2 ==: 175004       ;KXT DPR REGISTER 2 WORD LOC.
175004          KL.SC2 ==: 175004       ;" " " " LOW BYTE LOC.
175005          KH.SC2 ==: 175005       ;" " " " HIGH BYTE LOC.

;
; ERROR REPORT BIT DEFINITION OF REGISTER 2
;

000001          KX$F01 ==: 1            ;CSR TEST FAILED
000002          KX$F02 ==: 2            ;RAM " "
000004          KX$F03 ==: 4            ;ROM " "
000010          KX$F04 ==: 10           ;CPU " "
000020          KX$F05 ==: 20           ;REVENT " "
000040          KX$F06 ==: 40           ;SLU1 " "
000100          KX$F07 ==: 100          ;SLU2 " "
000200          KX$F10 ==: 200          ;PARALLEL I/O TEST FAILED
000400          KX$F11 ==: 400          ;DMA TEST FAILED
001000          KX$F12 ==: 1000         ;QIR " "
002000          KX$F13 ==: 2000         ;DPR " "

;
; DUAL PORT RAM SYSTEM CONTROL REGISTER 3
;

175006          KW.SC3 ==: 175006       ;KXT DPR REGISTER 3 WORD LOC.
175006          KL.SC3 ==: 175006       ;" " " " LOW BYTE LOC.
175007          KH.SC3 ==: 175007       ;" " " " HIGH BYTE LOC.

;
; Data Channel Base addresses (on the KXT side)
175010          KW.DCO ==: 175010       ; CHANNEL 0
175020          KW.DC1 ==: 175020       ; CHANNEL 1

;
; Offsets to Data Channel Base address
000000          KW.DCO ==: 0            ; Command word
000002          KW.DST ==: 2            ; Status word
000004          KW.DAT ==: 4            ; beginning of data

;
; Bit definitions within Data Channel Command
;

000036          KC.COM ==: 36           ; Command field
000040          KC.ICC ==: 40           ; Interrupt on command complete
000100          KC.IDA ==: 100          ; Interrupt on DA
000200          KC.IDR ==: 200          ; Interrupt on DR
003400          KC.LEN ==: 3400         ; Length field
004000          KC.EOM ==: 4000         ; End of message

```

```

177400          KC.VEC ==: 177400      ; Vector Number (high byte)

; Bit definitions within Data Channel Status
000036          KS.FRC ==: 36          ; Error code
000040          KS.DR ==: 40          ; Data requested (Write will succeed)
000100          KS.FOM ==: 100         ; End of message
000200          KS.DA ==: 200         ; Data available (Read will succeed)
003400          KS.ALN ==: 3400        ; Actual length of transfer
020000          KS.DBG ==: 20000       ; Debug available
040000          KS.ON ==: 40000        ; Interface ready (on)
100000          KS.ERR ==: 100000       ; Error (Cumulative)

; Data Channel Command Codes
;
000000          KC$NOP ==: 0.          ; NOP command
000002          KC$RES ==: 2.          ; Reset command
000004          KC$EI ==: 4.          ; Enable Interrupts command
000006          KC$DI ==: 6.          ; Disable Interrupts command
000010          KC$GS ==: 8.          ; Get Status command
000012          KC$SS ==: 10.         ; Set Status command
000014          KC$RD ==: 12.         ; Read data command
000016          KC$WD ==: 14.         ; Write data command
000020          KC$FD ==: 16.         ; Enable Debug command
000022          KC$DD ==: 18.         ; Disable Debug command
000022          KC$MAX ==: KC$DD       ; Maximum command code

; Data Channel Error Codes
;
000000          KE$OK ==: 0.          ; Success (no)error code
000002          KE$NDA ==: 2.          ; No data available (read rejected)
000004          KE$NDR ==: 4.          ; No data requested (write rejected)
000006          KE$ILC ==: 6.          ; Illegal command field
000010          KE$ILL ==: 8.          ; Illegal length field
000012          KE$ILV ==: 10.         ; Illegal vector
000014          KE$DNA ==: 12.         ; Debug not available
000014          KE$MAX ==: KE$DNA     ; Maximum error code

; .nlist me
;
```

KXTNF - KXT11CA Native Firmware MACRO V05.00 00:51 Page 4
NFDEF - Native Firmware definitions

```
1          .SBTTL NFDEF - Native Firmware definitions
2          .ENABLE LC,GBL
3
4          ;
5          ; Module name: DEF - DEFINITIONS
6          ;
7          ; System: KXT11-CA Native Firmware
8          ;
9          ;
10         ;
11         ;
12         ;
13         ;
14         ; Functional Description:
15         ;
16         ;
17         ; This module defines addresses and bit definitions for various I/O registers
18         ; on the KXT11-CA.
19         ;
```

```
1 ; BIT SYMBOL DEFINITIONS
2 ;
3 ;
4 000001      BIT0 == 1
5 000002      BIT1 == 2
6 000004      BIT2 == 4
7 000010      BIT3 == 10
8 000020      BIT4 == 20
9 000040      BIT5 == 40
10 000100     BIT6 == 100
11 000200     BIT7 == 200
12 000400     BIT8 == 400
13 001000     BIT9 == 1000
14 002000     BIT10 == 2000
15 004000     BIT11 == 4000
16 010000     BIT12 == 10000
17 020000     BIT13 == 20000
18 040000     BIT14 == 40000
19 100000     BIT15 == 100000
20 ;
21 ; SLU1 ADDRESS DEFINITIONS
22 ;
23 177560     RCSRIA == 177560
24
25 004000     RCVACB == BIT11
26 000200     RCVDNB == BIT7
27 000100     RCVIEB == BIT6
28
29 177562     RBUFIA == 177562
30
31 100000     ERRB == BIT15
32 040000     ORERRB == BIT14
33 020000     FRERRR == BIT13
34 004000     RCVAKB == BIT11
35
36 177564     XCSR1A == 177564
37
38 000200     XRDYB == BIT7
39 000100     XIEB == BIT6
40 000070     PBRF == BIT5!BIT4!BIT3
41 000004     MAINTB == BIT2
42 000002     PBREQ == BIT1
43 000001     XRRKB == BIT0
44
45 177566     XBUFIA == 177566
46
47 ;
48 ; SLU2 Definitions
49 ;
50 175700     SL2SAA == 175700      ;SLU 2 channel A Status register address
51 175710     SL2SBA == 175710      ;" " " " B " " "
52 175704     SL2CAA == 175704      ;" " " " A Command register   "
53 175714     SL2CBA == 175714      ;" " " " B " " "
54 ;
55 ; PIO Definitions
56 ;
57 177000     PIUICA == 177000      ;PIO interrupt control register address
```

KXTMF - KXT11CA Native Firmware MACRO V05.00 00:51 Page 5-1
NFDEF - Native Firmware definitions

58	177004	PTOAVA == 177004	;PIO Port A interrupt vector register
59	177006	PIOBVA == 177006	; " " B " " "
60	177010	PIOTVA == 177010	; " Counter Timer " " "
61		;	
62		; DMA Definitions	
63		;	
64	174470	DHAMMA == 174470	;DMA Master Mode Register
65	174532	DMAOVA == 174532	;DMA Channel 0 Interrupt Vector Register
66	174530	DMA1VA == 174530	; " " 1 " " "
67	174442	DMA0OA == 174442	;DMA Channel 0 Chain Offset Field
68	174446	DMA0SA == 174446	; " " " " Segment/Tag Field
69	174454	DMACDA == 174454	; " Command Register
70		;	
71		;	
72		; 8255 Control Register	
73		;	
74	177526	C8255 == 177526	
75			

```
1 .SBTTL SELF TEST OVERLAY 0
2 .LIST ME,MEB,SEQ,LOC,BIN ; NORMAL LISTING MODE
3 .NLIST MC,CND,REX ; DITTO
4 .FABL AMA,GBL
5 160000 . = 160000
6 ;
7 ; Module name: SELF TEST OVERLAY 0
8 ;
9 ; System: KXT11-CA Native Firmware
10 ;
11 ;
12 ;
13 ;
14 ;
15 ;
16 ; Functional Description:
17 ;
18 ; This module contains overlay 0 of the self test code. The self test
19 ; modules in this overlay are:
20 ;
21 ; I/O register check,
22 ; Native (and user) RAM,
23 ; Native (and user) ROM,
24 ; CPU (a null test),
25 ; Line clock (BEVNT) interrupt,
26 ; Console serial port (DC319), and
27 ; Second serial port (NEC7201).
28 ;
29 ; Overlay 0 physically occupies the first 2 KB of the boot ROM. It
30 ; executes starting at 160000 when the DIAG PROM EN bit, bit 5, in
31 ; KXTCSPA is 0.
32 ;
33 ; RETURNED ERRORS ARE BIT ENCODED IN R0 (TTEEEE)
34 ; WHERE T = TEST NUMBER
35 ; . AND E = 12 DISCRETE ERROR FLAGS.
36 ; . I.E. 010004 = TEST 01, ERROR(BIT) 2.
```

```
1 ; SCRATCH RAM ALLOCATION.
2 ;
3 ; R5 IS ODT'S "TOP-OF-MEM" POINTER, AND IS PRESERVED AS SUCH.
4 ; ALL SCRATCH REFERENCES ARE INDEXED FROM R5 AS FOLLOWS:
5 ;
6 ; ; LO MID HI
7 ; ; -----
8 000005 $TOP= $5 ; 077776 137776 157776
9 ;
10 150002 BUFR1= -27776 ; 050000 110000 130000 TOP-6KW
11 160002 BUFR2= -17776 ; 060000 120000 140000 TOP-4KW
12 170002 TMP1= -07776 ; 070000 130000 150000 TOP-2KW
13 170004 TMP2= TMP1+2 ;
14 170006 TMP3= TMP1+4 ;
15 170010 TMP4= TMP1+6 ;
16 170012 TMP5= TMP1+10 ;
17 170014 TMP6= TMP1+12 ;
18 170016 TMP7= TMP1+14 ;
19 170020 TMP8= TMP1+16 ;
20 177776 SEF1= -2 ; LAST 2 WORDS OF NATIVE RAM (BATTERY-BACK-FLAGS)...
21 000005 SEF= $TOP ; ...ARE UTILIZED AS OUR ERROR FLAGS.
22 ;
23 ; MISCELLANEOUS STUFF,
24 ;
25 000000 STN= 0 ; INIT TEST NUMBER SEQUENCE.
26 ;
27 ; A FEW MACROS.
28 ;
29 .MACRO BEGIN NAME,TAG,V1,V2,V3
30 .NLST
31 STN=$TN+1 ; SET TEST NUMBER...
32 SFN=0 ;...AND RESET ERROR NUMBER.
33 .LIST
34 .IRP N,\$TN
35 .SBttl - T^N NAME
36 ;*****
37 ;** TEST N -- NAME
38 ;*****
39 TST^N:
40 TAG:
41 .ENDR
42 .IF NB <V1>
43 MOV V1,-(SP) ; SAVE "V1".
44 MOV V1+2,-(SP)
45 .IF NB <V2>
46 MOV V2,-(SP) ; SAVE "V2".
47 MOV V2+2,-(SP)
48 .IF NB <V3>
49 MOV V3,-(SP) ; SAVE "V3".
50 MOV V3+2,-(SP)
51 .ENDC
52 .ENDC
53 .ENDC
54 CLR ($EF) ; CLEAR ERROR FLAG.
55 .ENDM BEGIN
56
57 .MACRO EXIT V1,V2,V3,?TAG
```

```
58          .IF NB <V1>
59             MOV      (SP)+,V1+2    ;; RESTORE "V1".
60             MOV      (SP)+,V1
61          .IF NB <V2>
62             MOV      (SP)+,V2+2    ;; RESTORE "V2".
63             MOV      (SP)+,V2
64          .IF NB <V3>
65             MOV      (SP)+,V3+2    ;; RESTORE "V3".
66             MOV      (SP)+,V3
67          .ENDC
68          .ENDC
69          .ENDC
70             MOV      (SEF),R0    ;; ERROR BITS => R0<11:00>...
71             BEQ      TAG      ;;...AND SKIP IF NONE.
72             .IRP      TN,\$TN
73                 BIS      #<TN*BIT12>,R0  ;; ELSE, ADD TEST NUM => R0<15:12>...
74             .ENDR
75             RETURN
76          .ENDM      EXIT      ;;...AND RETURN (NZ).
77
78          .MACRO  ERROR  TEXT
79             .NLIST  ME
80             .RADIX 10
81             .IRP      N,\$EN
82                 BTS      #BIT*N,($FF)   ;; E*N = "TEXT"
83             .FNDR
84                 SEN=$EN+1
85             .RADIX 8
86             .NLIST
87                 .LIST  MF
88                 .LIST
89             .ENDM      ERROR
90
91          .MACRO  LRESET
92             S0B      R0,.
93                 RESET
94                 S0B      R0,.
95             .ENDM      LRESET      ;; LOCAL RESET.
96
97             .NLIST  MD
98
```

16

```
1           ; A FEW DEFINITIONS
2           ;
3           ; STANDARD STUFF.
4           ;
5           000000    PRO= 000      ; THE PRIORITIES.
6           000040    PR1= 040
7           000100    PR2= 100
8           000140    PR3= 140
9           000200    PR4= 200
10          000240   PR5= 240
11          000300   PR6= 300
12          000340   PR7= 340
13          ;
14          ; PRIMARY CONTROL REGISTERS.
15          ;
16          177560    $SL1= 177560    ; DEC DC319 (DLART) ASYNC SERIAL PORT.
17          175700    $SL2= 175700    ; NEC7201 SYNC/ASYNC SERIAL PORT.
18          175720    $I8254= 175720  ; I8254 PROGRAMMABLE BAUD GEN FOR ABOVE.
19          177000    $PIO= 177000   ; Z8036 PIO/PIT.
20          174400    $DMA= 174400   ; AM28016 DMA CONTROLLER.
21          177520    $CSRA= 177520  ;
22          177522    $CSRB= 177522  ; > SECONDARY CSR'S...
23          177524    $CSRC= 177524  ; / ...EMBEDDED IN I8255 CHIP.
24          177526    $CSRCON= 177526 ;
25          177530    $CSR= 177530   ; PRIMARY CSR.
26          177532    $QIR= 177532   ; Q-BUS INTERRUPT REGISTER.
27          175000    $DPR= 175000   ; DUAL-PORT RAM (LOCAL).
28          160000    $QDPR1= 160000  ; DUAL-PORT RAM (GLOBAL BASE ID 0-7...).
29          175400    $QDPR2= 175400  ; ...AND ID 8-F).
30          ;
31          175000    $IPV= $DPR     ; IPV INTERRUPTS (VIA DPR WORDS 0, 4, 8, AND 12).
32          177524    $LEDS= $CSRC    ; CSRC<3:0> DRIVE THE LEDS.
33          177524    $UID= $CSRC    ; CSRC<7:4> SHOW THE ID SWITCH SETTING.
```

1 ; DEFAULT VECTORS.
2 ;
3 000024 PWRV= 024 ; PWR-UP (BRESET) (NON-MASKABLE).
4 000060 SL1RV= 060 ; CONSOLE (SLU1) RCVR (PRI 4).
5 000064 SL1XV= 064 ; CONSOLE (SLU1) XMTR (PRI 4).
6 000070 NECV= 070 ; SYNC/ASYNC BOTH CHANS (PKI 4).
7 000100 BEVNT= 100 ; LINE CLOCK (PRI 6).
8 000104 PEVNT= 104 ; PROGRAMMABLE CLOCK(S) (PRI 6).
9 ; 110 ; OPEN
10 000114 PTYV= 114 ; MEMORY PARITY (PRI 7).
11 000120 DPRV4= 120 ; DUAL-PORT RAM WORD 4 (PRI 5).
12 000124 DPRV8= 124 ; DUAL-PORT RAM WORD 8 (PKI 5).
13 000130 BIACKV= 130 ; Q-BUS "IACK" (PRI 5).
14 000134 DPRV12= 134 ; DUAL-PORT-RAM WORD 12 (PRI 5).
15 ; 140 ; PRESERVED (FALCON).
16 000144 QIRV= 144 ; Q-BUS REQUEST (ARBITER EXECUTES RTI).
17 000150 QIRV1= 150 ; DITTO (ARBITER EXECUTES RESET,RTI)
18 000154 PIOAV= 154 ; PIO PORT A (PRI 4).
19 000160 PIOBV= 160 ; PIO PORT B (PRI 4).
20 000164 DMAV= 164 ; DMA (BOTH CHANNELS) (PRI 4).

KXTNF - KXT11CA Native Firmware MACRO V05.00 00:51 Page 10
SELF TEST OVERLAY 0

```
1 ; LOW SEGMENT ENTRY BLOCK
2 ;
3 ; CALL: BIC #BITS5,CSRA ; ENABLE LO SEGMENT.
4 ; CALL TN ; CALL TEST 0-7.
5 ; BEQ OK ; OR BNE ERROR.
6 ;
7
8 14      160000
9     160000 005000
10    160002 000207
11    160004
12    160004
13    160004 000137 160040
14    160010
15    160010 000137 160272
16    160014
17    160014 000137 160700
18    160020
19    160020 000137 161150
20    160024
21    160024 000137 161164
22    160030
23    160030 000137 161344
24    160034
25    160034 000137 161730
26
27    T0==. CLR   R0      ;SINCE THERE IS NO TEST BY THIS NAME, SHOW NO ERRORS.
28    T1==. RETURN
29    T2==. JMP   TST1      ;; EXECUTE TEST 1.
30    T3==. JMP   TST2      ;; EXECUTE TEST 2.
31    T4==. JMP   TST3      ;; EXECUTE TEST 3.
32    T5==. JMP   TST4      ;; EXECUTE TEST 4.
33    T6==. JMP   TST5      ;; EXECUTE TEST 5.
34    T7==. JMP   TST6      ;; EXECUTE TEST 6.
35    T8==. JMP   TST7      ;; EXECUTE TEST 7.
```

```

1          .SBTTL - T1      I/O REGISTER CHECK
2          ;***** TEST 1 -- I/O REGISTER CHECK *****
3          ;***** TEST 1 -- I/O REGISTER CHECK *****
4
5          160040
6          160040
7          160040  005015
8          TST1:
9          TIOR:    CLR      ($SEF)           ;; CLEAR ERROR FLAG.
10         ;
11         ; INPUT:  NONE.
12         ; OUTPUT: R0 = ERROR FLAGS OR ZERO.
13         ;
14         ; VERIFY THAT ALL ASSIGNED I/O ADDRESSES ARE VALID.
15         ;
16         ;          T1      I/O REGISTER CHECK
17         ;          .
18         ;          E0 = BUS-ERROR AT CSR ADDRESS
19         ;          E1 = BUS-ERROR AT QIR ADDRESS
20         ;          E2 = BUS-ERROR AT DPR ADDRESS
21         ;          E3 = BUS-ERROR AT DC319 ADDRESS
22         ;          E4 = BUS-ERROR AT NEC7201 ADDRESS
23         ;          E5 = BUS-ERROR AT I8254 ADDRESS
24         ;          E6 = BUS-ERROR AT Z8036 ADDRESS
25         ;          E7 = BUS-ERPOP AT Z8016 ADDRESS
26
27
28 160042  012700  177520   1$:  MOV      #SCSRA,R0      ; THE CSR/SET...
29 160046  012701  000005   MOV      #5.,R1      ;...5 REGISTERS.
30 160052  004737  160254   CALL    10$      BIS      #R1TO,($SEF)  ;; E0 = BUS-ERROR AT CSR ADDRESS
31 160056  052715  000001
32
33 160062  012700  177532   2$:  MOV      #SQIR,R0      ; THE QIR...
34 160066  012701  000001   MOV      #1.,R1      ;...ONE-OF-A-KIND.
35 160072  004737  160254   CALL    10$      BIS      #BIT1,($SEF)  ;; E1 = BUS-ERROR AT QIR ADDRESS
36 160076  052715  000002
37
38 160102  012700  175000   3$:  MOV      #SDPR,R0      ; THE 2-PORT-RAM...
39 160106  012701  000020   MOV      #16.,R1      ;...16 REGISTERS.
40 160112  004737  160254   CALL    10$      BIS      #BIT2,($SEF)  ;; E2 = BUS-ERROR AT DPR ADDRESS
41 160116  052715  000004
42
43 160122  012700  177560   4$:  MOV      #SSL1,R0      ; THE CONSOLE PORT...
44 160126  012701  000004   MOV      #4.,R1      ;...4 REGISTERS.
45 160132  004737  160254   CALL    10$      BIS      #BIT3,($SEF)  ;; E3 = BUS-ERROR AT DC319 ADDRESS
46 160136  052715  000010
47
48 160142  012700  175700   5$:  MOV      #SSL2,R0      ; THE SYNC/ASYNC PORT...
49 160146  012701  000010   MOV      #8.,R1      ;...8 REGISTERS...
50 160152  004737  160254   CALL    10$      BIS      #BIT4,($SEF)  ;; E4 = BUS-ERROR AT NEC7201 ADDRESS
51 160156  052715  000020
52
53 160162  012700  175720   6$:  MOV      #I8254,R0      ;...AND ITS CLOCK GENERATOR...
54 160166  012701  000010   MOV      #8.,R1      ;...HAS 8 MORE.
55 160172  004737  160254   CALL    10$      BIS      #BIT5,($SEF)  ;; E5 = BUS-ERROR AT I8254 ADDRESS
56 160176  052715  000040
57
58 160202  012700  177000   7$:  MOV      #SPIO,R0      ; THE PARALLEL I/O PORT...
59 160206  012701  000060   MOV      #48.,R1      ;...A WHOPPING 48 REGISTERS.
60 160212  004737  160254   CALL    10$      BIS      #BIT6,($SEF)  ;; E6 = BUS-ERROR AT Z8036 ADDRESS
61 160216  052715  000100

```

KXINF - KXT11CA Native Firmware MACRO V05.00 00:51 Page 11-1
- T1 I/O REGISTER CHECK

```
52
53 160222 012700 174400      8$:    MOV    #$SDMA,R0      ; THE DMA ENGINE...
54 160226 012701 000056      MOV    #46.,R1      ;...A CLOSE SECOND WITH 46.
55 160232 004737 160254      CALL   10$      
56 160236 052715 000200      BIS    #BIT7,(SEF)    ;; E7 = BUS-ERROR AT Z8016 ADDRESS
57 160242              9$:    MOV    (SEF),R0      ;; ERROR BITS => R0<11:00>...
160242 011500              BEQ    30000$      ;;...AND SKIP IF NONE.
160244 001402              BIS    #<1*BIT12>,R0    ;; ELSE, ADD TEST NUM => R0<15:12>...
160246 052700 010000      30000$: RETURN      ;;...AND RETURN (NZ).
160252 000207              58
59 160254 005720      10$:   TST    (R0)+      ; TEST AN ADDRESS.
60 160256 000240      NOP    
61 160260 103403      BCS    11$      
62 160262 077104      S0B    R1,10$      ; TAKE ERROR RETURN IF IT TRAPPED.
63 160264 062716 000004      ADD    #4,(SP)      ; ALL OK, TAKE SKIP RETURN.
64 160270 000207      11$:   RETURN      ;
```

21

```

1           .SBTTL - T2      NATIVE (AND USER) RAM
2           ;;*****TEST 2 -- NATIVE (AND USER) RAM*****
3           ;;*****TEST 2 -- NATIVE (AND USER) RAM*****
4           ;;*****TEST 2 -- NATIVE (AND USER) RAM*****
5           160272          TST2:
6           160272          TRAM:
7           160272 005015      CLR      ($EF)        ;; CLEAR ERROR FLAG.
8
9           ; INPUT: R1 = NONE.
10          ; OUTPUT: R0 = ERROR FLAGS OR ZERO.
11          ;
12          ; TEST LOCAL RAM FROM BOTTOM UP (NON-DESTRUCTIVE).
13          ; IF SOCKET "A" CONTAINS USER RAM, TEST IT AS WELL.
14          ; 1. WRITE (DATO) EACH LOCATION WITH IT'S OWN ADDRESS.
15          ; 2. READ (DATI) AND VERIFY CORRECT DATA.
16          ; 3. COMPLIMENT WORD (DATIO) AND VERIFY.
17          ; 4. COMPLIMENT LO AND HI BYTES (DATIO(B)) AND VERIFY.
18          ;
19          ; T2      NATIVE (AND USER) RAM
20          ; E0 = BUS-ERROR AT RAM ADDRESS
21          ; .      E1 = WRITE-READ ERROR
22          ; .      E2 = READ-MOD-WRITE ERROR
23          ; .      E3 = READ-MOD-WRITE(LB) ERROR
24          ; .      E4 = READ-MOD-WRITE(HB) ERROR
25          ; .      E5 TO E9 = SAME AS E0 TO E4 IN USER RAM SPACE
26
27          22 160274 020527 137776      CMP      STOP,#137776
28          23 160300 101007      BMI      MEM3          ; BP IF RAM IS IN HIGH...
29          24 160302 001403      BEQ      MEM2          ;...OR MID RANGE.
30          25 160304 012702 000000      MEM1:   MOV      #0,R2          ; SET BOTTOM AT 0.
31          26 160310 000405      BR      MEM3A
32          27 160312 012702 040000      MEM2:   MOV      #40000,R2          ; SET BOTTOM AT 8K.
33          28 160316 000402      BR      MEM3A
34          29 160320 012702 100000      MEM3:   MOV      #100000,R2          ; SET BOTTOM AT 16K.
35          30 160324 010504      MEM3A:  MOV      STOP,R4          ; SET TOP.
36          31 160326 005046      CLR      -(SP)          ; CLEAR NATIVE/USER SWITCH.
37          32 160330 000241      1$:    CLC
38          33 160332 011200      MOV      (P2),R0          ; SAVE TARGET DATA.
39          34 160334 000240      NOP
40          35 160336 103003      BCC      2$              ; E0 = BUS-ERROR AT RAM ADDRESS
41          36 160340 052715 000001      BIS      #BIT0,($EF)
42          37 160344 000444      BR      7$              ; E0 = BUS-ERROR AT RAM ADDRESS
43
44          38 160346 010201      2$:    MOV      R2,R1          ; TARGET ADDRESS => R1.
45          39 160350 010211      MOV      R2,(R1)          ; WRITE (DATO)...
46          40 160352 011103      MOV      (R1),R3          ;...READ (DATI)...
47          41 160354 020203      CMP      R2,R3          ;...AND CHECK.
48          42 160356 001402      BREQ   3$              ; E1 = WRITE-READ ERROR
49          43 160360 052715 000002      BIS      #BIT1,($EF)
50          44 160364 005102      3$:    COM      R2
51          45 160366 005111      COM      (R1)          ; READ-MOD-WRITE (DATIO)...
52          46 160370 021102      CMP      (R1),R2          ;...READ AND CHECK.
53          47 160372 001402      BEQ      4$              ; E2 = READ-MOD-WRITE ERROR
54          48 160374 052715 000004      BIS      #BIT2,($EF)
55          49 160400 005102      COMB   (R1)          ; READ-MOD-WRITE-LB (DATIO(LB))...
56          50 160402 105111      COMB   (R1)
```

```

52 160404 121102 CMPB (R1),R2 ;...READ AND CHECK.
53 160406 001402 BFQ $S
54 160410 052715 000010 BIS #BIT3,$(SEF) ; E3 = READ-MOD-WRITE(LB) ERROR
55 160414 000302 SWAB R2
56 160416 105161 000001 COMB 1(R1) ; READ-MOD-WRITE-HB (DATIO(HB))...
57 160422 126102 000001 CMPB 1(R1),R2 ;...READ AND CHECK.
58 160426 001402 BFQ $S
59 160430 052715 000020 BIS #BIT4,$(SEF) ; E4 = READ-MOD-WRITE(HB) ERROR
60
61 160434 000302 SWAB R2
62 160436 010022 MOV R0,(R2)+ ; RESTORE AND BUMP TARGET ADDRESS.
63 160440 020227 160330 CMP R2,#1$ ; **** IF RUNNING IN RAM, DON'T TEST THE "TEST".
64 160444 001002 BNE 65$ ; ****
65 160446 012702 160456 MOV #7$,R2 ; ****
66 160452 020204 CMP R2,R4 ; REACHED TOP ??
67 160454 101725 BLOS 1$ ; LOOP IF NOT.
68
69 160456 005116 COM (SP) ; TOGGLE NATIVE/USER SWITCH...
70 160460 001416 BFQ $S ;...AND EXIT IF BOTH DONE.
71 160462 011565 177776 MOV $(SEF),-2$(SEF) ; SAVE NATIVE ERRORS...
72 160466 005015 CLR $(SEF) ;...AND CLEAR FOR 2ND PASS.
73 160470 004737 160546 CALL SKTA ; GET SOCKET SPECS.
74 160474 020227 000003 CMP R2,#3 ; CHECK WHAT'S THERE...
75 160500 001406 BFQ $S ;...AND BR IF ROM (OR EMPTY).
76 160502 010002 MOV R0,R2 ; IT'S RAM, 1ST ADDRESS => R2...
77 160504 010104 MOV R1,R4 ;...AND SIZE => R4.
78 160506 005304 DEC R4 ; \
79 160510 006304 ASL R4 ; > CHANGE SIZE TO LAST ADDRESS...
80 160512 060204 ADD R2,R4 ; /
81 160514 000705 BR 1$ ;...AND GO 'ROUND ONCE MORE.
82
83 ; E5 TO E9 = SAME AS E0..TO E4 IN USER RAM SPACE
84 ;
85 160516 005726 8$: TST (SP)+ ; DONE, POP THE STACK.
86 160520 012700 000005 MOV #5,R0
87 160524 006315 9$: ASL $(SEF) ; SHIFT USER ERRORS TO BITS<9:5>...
88 160526 077002 SDB R0,9$ ;...AND ADD NATIVE ERRORS IN BITS<4:0>.
89 160530 056515 177776 BTS -2$(SEF),(SEF) ; ERROR BITS => R0<11:00>...
90 160534 011500 MOV $(SEF),R0 ;...AND SKIP IF NONE.
160536 001402 BEQ 30001$ ; ELSE, ADD TEST NUM => R0<15:12>...
160540 052700 020000 BIS #<2*BIT12>,R0 ;...AND RETURN (NZ).
160544 000207 30001$: RETURN

```

```

1 ; SUBROUTINE TO DETERMINE USER (SOCKET A) ADDRESS AND SIZE.
2 ;
3 ; RETURN R0 = 1ST ADDRESS
4 ; P1 = SIZE (WORDS)
5 ; R2 = 3 IF ROM (OR EMPTY).
6 ;
7
8 160546 012700 100000 SKTA: MOV #100000,R0 ; ASSUME ROM OVER RAM.
9 160552 005705 TST $TOP
10 160554 100001 BPL 1$ ; BR IF SO...
11 160556 005000 CLR R0 ;...ELSE, SET ROM AT 0.
12 160560 012701 004000 177522 1$: MOV #2048.,R1 ; START SIZING AT 2KW.
13 160564 032737 000006 177522 BIT #<3*BIT1>,$CSR8
14 160572 001417 BEQ 2$ ; BR IF 2KW SELECTED (MAP 0 OR 4).
15 160574 006301 ASL R1 ; RAISE TO 4KW.
16 160576 032737 000004 177522 BIT #<2*BIT1>,$CSR8
17 160604 001412 BEQ 2$ ; BR IF 4KW SELECTED (MAP 1 OR 5).
18 160606 006301 ASL R1 ; RAISE TO 8KW.
19 160610 032737 000010 177522 BIT #<4*BIT1>,$CSR8
20 160616 001405 BEQ 2$ ; BR IF 8KW SELECTED (MAP 2 OR 3).
21 160620 032737 000002 177522 BIT #<1*BIT1>,$CSR8
22 160626 001401 BEQ 2$ ; DITTO (MAP 6).
23 160630 006301 ASL R1 ; RAISE TO 16KW (MAP 7).
24 160632 000401 2$: BR RAMLOA ; EXIT THRU ROM/RAM CHECKER.

25 ;
26 ; SUBROUTINE TO ESTABLISH ROM OR RAM IN VECTOR SPACE.
27 ;
28 ; RETURN R2<00> = 1 IF LO BYTE NON-WRITABLE.
29 ; R2<01> = 1 IF HI BYTE NON-WRITABLE.
30 ;
31 160634 005000 RAMLO: CLR R0
32 160636 005002 RAMLOA: CLR R2
33 160640 011046 MOV (R0),-(SP) ; GET TARGET LOCATION...
34 160642 005116 COM (SP) ;...AND COMPLIMENT IT.
35 160644 005110 COM (R0) ; COMPLIMENT TARGET LOCATION.
36 160646 126066 000001 000001 CMPB 1(R0),1(SP) ; HI PYTE WRITABLE ??
37 160654 001402 BEQ 1$ ; YES
38 160656 052702 000002 BIS #BIT1,R2 ; NO
39 160662 121026 1$: CMPB (R0),(SP)+ ; LO BYTE WRITABLE ??
40 160664 001402 BEQ 2$ ; YES
41 160666 052702 000001 BIS #BIT0,R2 ; NO
42 160672 005110 2$: COM (R0) ; RESTORE TARGET LOCATION.
43 160674 005702 TST R2 ; SET CONDITION FOR CALLER.
44 160676 000207 RETURN

```

```

1           .SBTTL - T3  NATIVE (AND USER) ROM
2           ;*****
3           ;** TEST 3 -- NATIVE (AND USER) ROM
4           ;*****
5
6           160700          TST3:
7           160700          TR0M:   CLR      (SEF)      ;; CLEAR ERROR FLAG.
8
9           160700 005015
10          T3      NATIVE (AND USER) ROM
11          .          E0 = LO BYTE CHECK SUM ERROR (NATIVE)
12          .          E1 = HI BYTE CHECK SUM ERROR (NATIVE)
13          .          E2 = LO BYTE CHECK SUM ERROR (USER)
14          .          E3 = HI BYTE CHECK SUM ERROR (USER)
15 160702 010104          MOV      R1,R4      ; SAVE OPTION FLAG.
16 160704 012700 161150          MOV      #ACC+2,RO
17 160710 012701 000043          MOV      #<ACC+2-CKSUM>/2,R1
18 160714 014046          1S:     MOV      -(R0),-(SP) ; PUSH CHECK-SUM CODE ONTO THE STACK.
19 160716 077102          SOB      R1,1$      R1,1$ 
20
21 160720 012700 160000          MOV      #160000,RO
22 160724 012701 010000          MOV      #4096.,R1
23 160730 004766 000000          CALL     0(SP)      ; CHECK SUM (LO BYTE).
24 160734 052715 000001          BIS      #BIT0,($EF) ;; E0 = LO BYTE CHECK SUM ERROR (NATIVE)
25
26 160740 012700 160001          2S:     MOV      #160001,RO
27 160744 012701 010000          MOV      #4096.,R1
28 160750 004766 000000          CALL     0(SP)      ; CHECK SUM (HI BYTE).
29 160754 052715 000002          BIS      #BIT1,($EF) ;; E1 = HI BYTE CHECK SUM ERROR (NATIVE)
30 160760
31           ; IF ERROR, CHECK FOR OVERLAY PROBLEM ???
32
33 160760 032704 000001          4S:     BIT      #BIT0,R4      ; USER ROM SELECTED ??
34 160764 001417          BEQ      6$      ; EXIT IF NOT.
35 160766 004737 160546          CALL     SKTA      ; YES, GET SOCKET SPECS.
36 160772 010046          MOV      R0,-(SP) ; PUSH START ADDRESS...
37 160774 010146          MOV      R1,-(SP) ;...AND SIZE.
38 160776 004766 000004          CALL     4(SP)      ; CHECK SUM (LO BYTE).
39 161002 052715 000004          BIS      #BIT2,($EF) ;; E2 = LO BYTE CHECK SUM ERROR (USER)
40
41 161006 012601          5S:     MOV      (SP)+,R1      ; POP SIZE...
42 161010 012600          MOV      (SP)+,R0      ;...AND ADDRESS.
43 161012 005200          INC      R0
44 161014 004766 000000          CALL     0(SP)      ; CHECK SUM (HI BYTE).
45 161020 052715 000010          BIS      #BIT3,($EF) ;; E3 = HI BYTE CHECK SUM ERROR (USER)
46
47 161024 062706 000106          6S:     ADD      #ACC+2-CKSUM,SP ; FIX STACK.
48 161030 011500          MOV      ($EF),R0      ;; ERROR BITS => R0<11:00>...
        161032 001402          BEQ      30002$      ;...AND SKIP IF NONE.
        161034 052700 030000          BIS      #<3*BIT12>,R0      ;; ELSE, ADD TEST NUM => R0<15:12>...
        161040 000207          30002$: RETURN      ;...AND RETURN (NZ).

```

```
1 ;*****  
2 ; ROM CHECK SUM ROUTINE.  
3 ; THIS STUFF IS COPIED TO AND EXECUTED IN STACK SPACE.  
4 ;  
5 .DSABL AMA ; RELATIVE PIC.  
6 ;  
7 ; ON ENTRY, R0 = START ADDRESS AND R1 = BYTE COUNT.  
8 ; ON RETURN, R2 = ACCUMULATED SUM AND R3 = CHECK-SUM.  
9 ;  
10 161042 005067 000100 CKSUM: CLR ACC ; CLEAR ACCUMULATOR...  
11 161046 005301 DEC R1 ;...AND DECR THE BYTE COUNT.  
12 161050 111002 1S: MOVB (F0),R2 ; 1ST/NEXT ROM LOCATION.  
13 161052 042702 177400 BIC #^C377,R2  
14 161056 060267 000064 ADD R2,ACC ; \  
15 161062 106367 000060 ASLB ACC ; > ACCUMULATE BYTE SUM.  
16 161066 005567 000054 ADC ACC ; /  
17 161072 062700 000002 ADD #2,R0  
18 161076 020027 174000 CMP R0,#174000 ; END OF NATIVE ROM ??  
19 161102 103405 BLO 2S ; BR IF NOT.  
20 161104 052737 000040 177520 BTS #BITS,0#$CSPA ; YES, MAP IN THE OVERLAY...  
21 161112 042700 017776 BIC #^C160001,R0 ;...AND ADJUST SRC POINTER.  
22 161116 077124 2S: SOB R1,1S ; LOOP 'TIL DONE.  
23 161120 116702 000022 MOVB ACC,R2 ; THEN PUT ACCUMULATED SUM IN R2...  
24 161124 111003 MOVB (R0),R3 ;...AND THE CHECK-SUM IN R3.  
25 161126 042737 000040 177520 BIC #BITS,0#$CSRA ; UNMAP OVERLAY.  
26 161134 120203 CMPB R2,R3 ; ACCUMULATED SUM SHOULD EQUAL CHECK-SUM.  
27 161136 001002 BNE 3S ; ERROR RETURN IF NOT.  
28 161140 062716 000004 ADD #4,(SP) ; SKIP RETURN IF SO.  
29 161144 000207 3S: RETURN  
30 161146 000000 ACC: 0  
31 .ENABL AMA  
32 ;*****
```

KXTNF - KXT11CA Native Firmware MACRO V05.00 00:51 Page 16
- T4 T11 CPU INSTRUCTIONS AND TRAPS

```
1          .SBTTL - T4      T11 CPU INSTRUCTIONS AND TRAPS
;*****T11 CPU INSTRUCTIONS AND TRAPS*****
;** TEST 4 -- T11 CPU INSTRUCTIONS AND TRAPS
;*****T11 CPU INSTRUCTIONS AND TRAPS*****
161150          TST4:
161150          TCPU:
161150 005015          CLR    ($EF)           ; CLEAR ERROR FLAG.
2
3          ; INPUT: NONE.
4          ; OUTPUT: R0 = ERROR FLAGS OR ZERO.
5
6          ; TEST THE CPU (WHATEVER THAT MEANS) !!
7
8          ;      T4      T11 CPU INSTRUCTIONS AND TRAPS
9          ;      .      NOT INSTALLED.
10
11 161152 011500          MOV    ($EF),R0      ; ERROR BITS => R0<11:00>...
161154 001402          BEQ    30003$      ;...AND SKIP IF NONE.
161156 052700 040000          BIS    #<4*BIT12>,R0      ; ELSE, ADD TEST NUM => R0<15:12>...
161162 000207          30003$: RETURN      ;...AND RETURN (N2).
```

```

1          .SETL - T5   LINE CLOCK (BEVNT) INTERRUPT
;***** TEST 5 -- LINE CLOCK (BEVNT) INTERRUPT *****
;***** TEST 5 -- LINE CLOCK (BEVNT) INTERRUPT *****
161164      TST5:
161164      TCLK:
161164 013746 000100      MOV    BEVNT,-(SP)    ; SAVE BEVNT.
161170 013746 000102      MOV    BEVNT+2,-(SP)  ;;
161174 005015      CLR    ($EF)     ; CLEAR ERROR FLAG.

2          ;
3          ; INPUT: NONE.
4          ; OUTPUT: R0 = ERROR FLAGS OR ZERO.
5          ;
6          ; LINE CLOCK IS JUMPERED FROM CONSOLE DC319 OR Q-BUS BEVNT.
7          ; VERIFY THAT WE CAN ENABLE/DISABLE THE CLOCK VIA CSRA<6>,
8          ; AND THAT IT INTERRUPTS THRU VECTOR 100 AT PRI 6.
9          ;
10         ; T5   LINE CLOCK (BEVNT) INTERRUPT
11         ; .   E0 = ROM IN VECTOR SPACE -- CAN'T RUN
12         ; .   E1 = CLOCK INTERRUPT NOT MASKED AT LEVEL 6
13         ; .   E2 = CLOCK DOESN'T INTERRUPT
14         ; .   E3 = CAN'T SHUT IT OFF

15
16 161176 004737 160634      CALL   RAMLO,        ; VECTOR SPACE USABLE ??
17 161202 001405      BEQ    2S        ; PROCEFD IF SO.
18 161204 052715 000001      BIS    #BIT0,$(EF)  ;; F0 = ROM IN VECTOR SPACE -- CAN'T RUN
19 161210 000444      BR    TCXIT
20
21 161212 005201      1S:    INC    R1        ; ON "BEVNT", TICK AND RETURN.
22 161214 000002      RTI
23
24 161216 106427 000300      2S:    MTPS   #PR6      ; RAISE CPU.
25 161222 012737 161212 000100      MOV    #1$,BEVNT  ; SET CLOCK VECTOR.
26 161230 012737 000300 000102      MOV    #PR6,BEVNT+2
27 161236 005000      CLR    R0        ; CLEAR LOOP CONTROL...
28 161240 005001      CLR    R1        ;...AND TICKER.
29 161242 052737 000100 177520      BIS    #BIT6,$CSRA  ; TURN ON CLOCK INTERRUPT.
30 161250 077001      SOB    R0,.     ; DELAY, INTERRUPT SHOULD...
31 161252 005701      TST    R1        ;...BE MASKED AT THIS LEVEL.
32 161254 001402      BEQ    3S        ; BR IF SO.
33 161256 052715 000002      BIS    #BIT1,$(EF)  ;; E1 = CLOCK INTERRUPT NOT MASKED AT LEVEL 6
34 161262 106427 000240      3S:    MTPS   #PR5      ; NOW, LOWER CPU TO PRI 5.
35 161266 077001      SOB    R0,.     ; DELAY AGAIN, CLOCK SHOULD COME IN...
36 161270 005301      DEC    R1        ;...AND GIVE US A FEW TICKS (MORE THEN 1).
37 161272 003002      BGT    4S        ; BR IF SO.
38 161274 052715 000004      BIS    #BIT2,$(EF)  ;; E2 = CLOCK DOESN'T INTERRUPT

39
40 161300 005037 177520      4S:    CLR    $CSRA     ; TURN IT OFF.
41 161304 000240      NOP
42 161306 005001      CLR    R1
43 161310 077001      SOB    R0,.
44 161312 005701      TST    R1     ;...CLOCK SHOULD BE SHUT OFF.
45 161314 001402      BEQ    6S     ; BR IF SO.
46 161316 052715 000010      BIS    #BIT3,$(EF)  ;; E3 = CAN'T SHUT IT OFF
47 161322      6S:TCXIT:
48 161322 012637 000102      MOV    (SP)+,BEVNT+2  ;; RESTORE BEVNT.
161326 012637 000100      MOV    (SP)+,BEVNT

```

KXTNF - KXT11CA Native Firmware MACRO V05.00 00:51 Page 17-1
- T5 LINE CLOCK (8EVNT) INTERRUPT

161332 011500	MOV	(\$EF),R0	;; ERROR BITS => R0<11:00>...
161334 001402	BEQ	30004\$;;...AND SKIP IF NONE.
161336 052700 050000	BIS	#<5*BIT12>,R0	;; ELSE, ADD TEST NUM => R0<15:12>...
161342 000207		30004\$: RETURN	;;...AND RETURN (NZ).

```

1          .SETTL - T6      CONSOLE SERIAL PORT DC319
;***** TEST 6 -- CONSOLE SERIAL PORT DC319
;*****
161344
161344
161344 013746 000064      TST6:
161350 013746 000066      TDC:
161354 013746 000060      MOV     SL1XV,-(SP)    ;> SAVE SL1XV.
161360 013746 000062      MOV     SL1XV+2,-(SP)
161364 005015              MOV     SL1RV,-(SP)    ;> SAVE SL1RV.
                                         MOV     SL1RV+2,-(SP)
                                         CLR     (SEF)      ;> CLEAR ERROR FLAG.

2
3          ; INPUT: NONE.
4          ; OUTPUT: R0 = ERROR FLAGS OR ZERO.
5
6          ; *** REQUIRES EXTERNAL LOOP-BACK ***
7
8          ; TEST THAT THE CONSOLE PORT WORKS AS ADVERTISED.
9
10         ; T6      CONSOLE SERIAL PORT DC319
11         ; .
12         ; E0 = ROM IN VECTOR SPACE -- INTERRUPTS NOT TESTED
13         ; E1 = XMTR INTERRUPT NOT MASKED AT LEVEL 4
14         ; E2 = XMTR INTERRUPT NOT RECEIVED
15         ; E3 = RCVR INTERRUPT NOT MASKED AT LEVEL 4
16         ; E4 = RCVR INTERRUPT NOT RECEIVED
17         ; E5 = RECEIVED DATA INCORRECT
18         ; E6 = NO RCVR DONE, LOOP-BACK OPEN
19 161366 012704 177560      MOV     #SSL1,R4      ; POINTFR => R4.
20 161372 004737 160634      CALL    RAMLO      ; VECTOR SPACE USABLE ???
21 161376 001403              BEQ    DC.XMT      ; PPOCEED IF SO.
22 161400 052715 000001      BTS    #BIT0,(SEF)  ;> E0 = ROM IN VECTOR SPACE -- INTERRUPTS NOT TESTED
23 161404 000503              BR     DC.DAT
24
25 161406 106427 000200      DC.XMT: MTPS    #PR4      ; RAISE CPU TO XMTR LEVEL.
26 161412 012737 161440 000064      MOV     #2$,SL1XV   ; SET VECTOR.
27 161420 012737 000200 000066      MOV     #PR4,SL1XV+2
28 161426 052764 000100 000004      BIS    #100,4(R4)  ; SET INTERRUPT ENABLE.
29 161434 000240              NOP
30 161436 000403              BR     3$        ; INTERRUPT SHOULD BE HELD OFF...
31 161440
32 161444 000411              2$:    BIS    #BIT1,(SEF)  ;> E1 = XMTR INTERRUPT NOT MASKED AT LEVEL 4
33 161446 012737 161470 000064 3$:    BR     4$        ; CHANGE THE VECTOR.
34 161454 106427 000000      MTPS    #PRO      ; LOWER CPU PRIORITY.
35 161460 000240              NOP
36 161462 052715 000004      BIS    #BIT2,(SEF)  ;> E2 = XMTR INTERRUPT NOT RECEIVED
37 161466 000401              BR     5$        ; INTERRUPT SHOULD HAVE COME IN.
38 161470 022626              CMP    (SP)+,(SP)+  ; ON INTERRUPT, FIX STACK.
39 161472 042764 000100 000004 4$:    BIC    #100,4(R4)  ; CLEAR THE INT ENABLE.
40
41 161500 016400 000002      DC.RCV: MOV     2(R4),R0      ; ENSURE RCVR DONE IS CLEAR.
42 161504 106427 000200      MTPS    #PR4      ; RAISE CPU LEVEL.
43 161510 005000              CLR     R0
44 161512 110064 000006      MOVB   R0,6(R4)   ;...TRANSMIT A NULL CHARACTER...
45 161516 105714              TSTR    (P4)      ;...WAIT 'TIL RCVR GETS IT...
46 161520 100402              BMI    2$
```

```

47 161522 077003 SOB R0,1$ ;...(BUT DON'T WAIT FOREVER).
48 161524 000462 BR DC.LBO
49
50 161526 012737 161552 000060 2$: MOV #3$,SL1RV ; OK, SET RCVR VECTOR.
51 161534 012737 000200 000062 MOV #PR4,SL1RV+2
52 161542 052714 000100 BIS #100,(R4) ; SET RCVR INTERRUPT ENABLE.
53 161546 000240 NOP ; INTERRUPT SHOULD BE HELD OFF...
54 161550 000403 BR 4$ ;...BR IF SO.
55 161552 161552 052715 000010 3$: BTS #BIT3,(SEF) ;; E3 = RCVR INTERRUPT NOT MASKED AT LEVEL 4
56 161556 000411 BR 5$ ;; AND LOWER CPU PRIORITY.
57 161560 012737 161602 000060 4$: MOV #5$,SL1PV ; CHANGE VECTOR...
58 161566 106427 000000 MTPS #PRO ;;...AND LOWER CPU PRIORITY.
59 161572 000240 NOP ; INTERRUPT SHOULD COME IN.
60 161574 052715 000020 BTS #BIT4,(SEF) ;; E4 = RCVR INTERRUPT NOT RECEIVED
61 161600 000401 BR 6$ ; ON INTERRUPT, FIX STACK.
62 161602 022626 5$: CMP (SP)+,(SP)+ ; CLEAR THE INT ENABLE...
63 161604 042714 000100 6$: BIC #100,(R4) ;;...AND LOWER RCVR DONE FLAG.
64 161610 016400 000002 MOV 2(R4),R0
65
66 161614 012700 163666 DC.DAT: MOV #FLT10,R0 ; GET DATA TABLE POINTER.
67 161620 005001 CLR R1
68 161622 005002 CLR R2
69 161624 010164 000006 10$: MOV R1,6(R4) ; XMIT A NULL...
70 161630 105714 TSTP (R4) ;;...AND SEE IF LOOP IS INTACT.
71 161632 100410 BMI 3$ ; IT IS, PROCEED AT 3$.
72 161634 077203 SOB R2,10$ ;; AND LOWER RCVR DONE FLAG.
73 161636 000415 BR DC.LBO ; IT'S NOT, ABORT.
74
75 161640 112001 1$: MOVR (R0)+,R1 ; NOW FLOAT DATA THRU THE LOOP...
76 161642 001415 BEQ DC.XIT ;...UNTIL DONE.
77 161644 010164 000005 MOV R1,6(R4) ; XMIT A BYTE...
78 161650 105714 '2$: TSTB (R4) ;;...AND WAIT.
79 161652 100376 BPL 2$ ; READ...
80 161654 016402 000002 3$: MOV 2(R4),R2 ;;...AND VERIFY.
81 161660 120102 CMPB R1,R2 ; LOOP IF OK.
82 161662 001766 BEQ 1$ ;; E5 = RECEIVED DATA INCORRECT
83 161664 052715 000040 BIS #BIT5,(SEF)
84 161670 000402 BR DC.XIT
85 161672 161672 052715 000100 DC.LBO: BIS #BIT6,(SEF) ;; E6 = NO RCVR DONE, LOOP-BACK OPEN
86 161676 161676 012637 000062 DC.XIT: MOV (SP)+,SL1RV+2 ;; RESTORE SL1RV.
161702 012637 000060 MOV (SP)+,SL1RV ;; RESTORE SL1XV.
161706 012637 000066 MOV (SF)+,SL1XV+2
161712 012637 000064 MOV (SP)+,SL1XV
161716 011500 MOV (SEF),R0 ;; ERROR BITS => R0<11:00>...
161720 001402 BEQ 30005$ ;;...AND SKIP IF NONE.
161722 052700 060000 BIS #<6*BIT12>,R0 ;; ELSE, ADD TEST NUM => R0<15:12>...
161726 000207 30005$: RETURN ;;...AND RETURN (NZ).

```

31

```

1           .SBTTL - T7   SECOND SERIAL PORT NEC7201
2           ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
3           ;;; TEST 7 -- SECOND SERIAL PORT NEC7201
4           ;;;*****
5
6           161730          TST7:
7           161730          TNEC:
8           161730          MOV     PEVNT,-(SP)      ;; SAVE PEVNT.
9           161734          MOV     PEVNT+2,-(SP)
10          161740          MOV     NECV,-(SP)      ;; SAVE NECV.
11          161744          MOV     NECV+2,-(SP)
12          161750          CLR     (SEF)          ;; CLRFAR ERROR FLAG.

13          ; INPUT: R1<BIT0> = RUN CHAN A ONLY (INCLUDES DMA AND MODEM CONTROL).
14          ;           R1<BIT1> = RUN CHAN B ONLY.
15          ; OUTPUT: R0 = ERROR FLAGS OR ZERO.
16
17          ; NEC7201 SYNC/ASYNC (MULTI-PROTOCOL) SERIAL INTERFACE.
18          ; INCLUDES AN INTEL 8254 TRIPLE PIT FOR BAUD RATE GENERATION.
19
20          ; FIRST, SET UP THE BAUD GENERATOR, AND VERIFY THAT
21          ; GENERAL PURPOSE TIMER 2 (800HZ) WORKS AS ADVERTISED.
22
23          ;       T7   SECOND SERIAL PORT NEC7201
24          ;       .
25          ;       E0 F ROM IN VECTOR SPACE -- CAN'T RUN
26          ;       E1 = 18254 TIMER 2 (800HZ) DOESN'T INTERRUPT
27          ;       E2 = ASYNC MODE, DATA XFER INCOMPLETE
28          ;       E3 = SYNC MODE, EOF-SDLC NOT RECEIVED
29          ;       E4 = SYNC MODE, DATA XFER INCOMPLETE
30          ;       E5 = SYNC/ASYNC MODES, RECEIVED DATA INCORRECT
31
32          21 161752 004737 160634      CALL   RAMLO      ; VECTOR SPACE USABLE ??
33          22 161756 001404      BFP   1$          ; PROCEED IF SO.
34          23 161760 052715 000001      BIS   #BIT0,(SEF)  ;; E0 = ROM IN VECTOR SPACE -- CAN'T RUN
35          24 161764 000137 163470      JMP   NECXIT
36
37          26 161770 012704 175736      1$:   MOV   #SSL2+36,R4  ; GET I8254 CONTROL REG ADDRESS.
38          27 161774 106427 000300      MTPS  #PR6        ; RAISE CPU.
39          28 162000 112714 000066      MOVB  #066,(R4)    ; CHAN A <TIMER-0!2-BYTES!MODE-3!BIN>.
40          29 162004 112714 000166      MOVB  #166,(R4)    ; CHAN B <TIMER-1! SAME >.
41          30 162010 112714 000264      MOVE  #264,(R4)    ; 800HZ <TIMER-2! SAME EXCEPT MODE-2>.
42          31 162014 013744 162124      MOVB  HZ800,-(R4)  ; SET 800HZ DIVIDER == 50HZ
43          32 162020 113714 162125      MOVB  HZ800+1,(R4)
44          33 162024 013744 162126      MOVB  A4800,-(R4)  ; SET CHAN B...
45          34 162030 113714 162127      MOVB  A4800+1,(R4)
46          35 162034 013744 162126      MOVB  A4800,-(R4)  ;...AND CHAN A FOR ASYNC 4800 (SYNC 76.8K).
47          36 162040 113714 162127      MOVR  A4800+1,(R4)
48
49          38 162044 012737 162120 000104      MOV   #3$,PEVNT  ; SET PEVNT VECTOR...
50          39 162052 012737 000300 000106      MOV   #PR6,PEVNT+2
51          40 162060 052737 000200 177520      BIS   #P1T7,SCSRA  ;...AND ENABLE IT.
52          41 162066 005002      CLR   R2
53          42 162070 005000      CLR   R0
54          43 162072 106400      MTPS  R0          ; LOWER CPU.
55          44 162074 077001      S0B   R0,-        ; DELAY, PEVNT SHOULD COME IN...
56          45 162076 005302      DEC   R2          ;...AND GIVE A FEW TICKS (MORE THAN 1).
57          46 162100 003002      BGT  2$          ;;
58          47 162102 052715 000002      BIS   #BIT1,(SEF)  ;; E1 = 18254 TIMER 2 (800HZ) DOESN'T INTERRUPT

```

KXTNF - KXT11CA Native Firmware MACRO V05.00 00:51 Page 19-1
- T7 SECOND SERIAL PORT NEC7201

```
48 162106 005037 177520      2$:    CLR    $CSRA      ; TURN OFF PEVNT...
49 162112 012714 000260      MOV    #260,(R4)   ;...AND TIMER 2 (MODE=0).
50 162116 000434             BR     NEC.DATA   ;...AND MOVE ON.
51
52 162120 005202             3$:    INC    R2        ; ON "PEVNT", TICK AND RETURN.
53 162122 000002             RTI
54
55 162124 000020             HZ800: 16.      ; DIVIDER YIELDS == 50HZ
56 162126 000200             A4800: 128.    ; DIVIDER YIELDS ASYNC 4800 AND/OR SYNC 76.8K.
```

```

1 ; FIVE PASS SYNC/ASYNC DATA TEST.
2 ;
3 ; *** REQUIRES EXTERNAL LOOP-BACKS (2) ***
4 ;
5 ; 1. CHAN B, ASYNC (4800 BAUD).
6 ; 2. CHAN A, ASYNC (4800 BAUD).
7 ; 3. CHAN B, SYNC (76.8K BAUD).
8 ; 4. CHAN A, SYNC (76.8K BAUD).
9 ; 5. CHAN A, SYNC, DMA DRIVEN.
10 ;
11 ;
12     000003      CHA= $3          ; DEDICATE R3 AS "CHAN-A" AND...
13     000004      CHX= $4          ;...R4 AS "CHAN-IN-TEST" PORT POINTERS.
14     177774      STAT= -4         ; INDICES TO CHAN A/B STATUS...
15     177776      RD8= -2         ;...RCVR DATA...
16     000000      CTRL= 0          ;...CONTROL...
17     000002      XDB= 2          ;...AND XMTR DATA PORTS.
18     170002      KAT= TMP1        ; EQUATE A KEEF-ALIVE-TIMER...
19     170004      MODE= TMP2       ;...SET-UP/MODE FLAG...
20     170006      STATS= TMP3       ;...STATUS BYTES...
21     170010      INHIB= TMP4       ;...AND INHIBIT (A/B) FLAG IN RAM.
22 ;
23 ; ASYNC SET-UP   R  CODE           FUNCTION
24 ;
25 162130 000 030      A.SET: .BYTE 0, 030 ; COMMAND <CHANNEL RESET>.
26 162132 002 024      .BYTE 2, 024 ; BUS INTERFACE <NON-VECTOREDIPRI1INO-DMA>.
27 162134 004 104      .BYTE 4, 104 ; PROTOCOL <X16IASYNC-1-STOPINO-PARITY>.
28 162136 003 301      .BYTE 3, 301 ; RCVR CONTROL <8-BPCIRX-ENAB>.
29 162140 005 152      .BYTE 5, 152 ; XMTR CONTROL <8-BPCITX-ENABIPRTYX-B>.
30 162142 000 020      .BYTE 0, 020 ; COMMAND <RESET-EXT-INTS>.
31 162144 001 036      .BYTE 1, 036 ; INT CONTROL <RXI-ON-ALLICAVITX-IEINO-EXT-IE>.
32 162146 177777      -1
33 ;
34 ; SYNC SET-UP   R  CODE           FUNCTION
35 ;
36 162150 000 030      S.SET: .BYTE 0, 030 ; COMAND <CHANNEL RESET>.
37 162152 002 024      .BYTE 2, 024 ; BUS INTERFACE <NON-VECTOREDIPRI1INO-DMA>.
38 162154 004 040      .BYTE 4, 040 ; PROTOCOL <X11SDLCISYNCINO-PARITY>.
39 162156 006 000      .BYTE 6, 000 ; SYNC 1 <SEARCH ADDRESS -- UNUSED>.
40 162160 007 176      .BYTE 7, 176 ; SYNC 2 <FLAG CODE>.
41 162162 003 311      .BYTE 3, 311 ; RCVR CONTROL <8-BPCICRC-ENABIRX-ENAB>.
42 162164 005 153      .BYTE 5, 153 ; XMTR CONTROL <8-BPCITX-ENABICCITTIPRTYX-BICRC-ENAB>.
43 162166 000 200      .BYTE 0, 200 ; COMMAND <RESET TX-CRC-GEN>.
44 162170 000 100      .BYTE 0, 100 ; COMMAND <RESET RX-CRC-GEN>.
45 162172 000 020      .BYTE 0, 020 ; COMMAND <RESET EXT-INT>.
46 162174 001 036      .BYTE 1, 036 ; INT CONTROL <RXI-ON-ALLICAVITX-IEINO-EXT-IE>.
47 162176 177777      -1
48 162200 001 004      C.CLR: .BYTE 1, 004 ; INT CONTROL <KEEP-CAVIDISABLE-ALL-BSF>
49 162202 003 300      .BYTE 3, 300 ; RCVR <RX-DISABLE>.
50 162204 005 142      .BYTE 5, 142 ; XMTP <TX-DISABLE>.
51 162206 177777      -1

```

34

```

1 ; FIRST, SEE IF EITHER CHAN IS INHIBITED.
2 ; NEC.DATA: CLR R0
3
4 162210 005000      ;#RIT1#BIT0,P1 ; SINGLE CHAN SELECTED ???
5 162212 032701 000003  BEQ 1S ; PROCEED IF NOT (RUN BOTH).
6 162216 001407      MOV #SSL2+14,R0 ; YES, ASSUME RUN A, INHIBIT B...
7 162220 012700 175714  BIT #RIT0,RI
8 162224 032701 000001  BNE 1S ;...AND SKIP IF WE GUESSED RIGHT.
9 162230 001002      MOV #SSL2+4,R0 ; ELSE, SET TO INHIBIT A.
10 162232 012700 175704 MOV R0,TNHIB(STOP) ; SET/CLEAR CHAN INHIBIT FLAG.
11 162236 010065 170010 1S: MOV R0,TNHIB(STOP) ; SET/CLEAR CHAN INHIBIT FLAG.
12
13 ; 1ST 4 PASSES.
14
15 162242 012700 162130  NEC.ASYNC: MOV #A.SET,R0 ; ASYNC MODE FIRST...
16 162246 000405      BR NEC.AA
17 162250 012700 162150  NEC.SYNC: MOV #S.SET,P0 ;...THEN, SYNC (SDLC) MODE...
18 162254 052737 000004 177520  RTS #4,SCSRA ;...REQUIRES <SYNCMA1-SYNCMB> IN CSRA.
19 162262 010065 170004  NEC.AA: MOV R0,MODE(STOP) ; SAVE POINTER AS SYNC/ASYNC FLAG.
20 162266 012703 175704  MOV #SSL2+4,CHA ; SET CHAN-A CONTROL PORT POINTER.
21 162272 012704 175714  MOV #SSL2+14,CHX ; SET CHAN-B AS CURRENT CHAN-IN-TEST.
22 162276 111013      MOV#P (R0),(CHA)
23 162300 116013 000001  MOV#B 1(R0),(CHA) ; COMMAND <CHAN-A-RESET (INCLUDES INT LOGIC)>.
24 162304 000240      NOP ; WAIT A MOMENT.
25 162306 112014      MOV#B (R0)+,(CHX)
26 162310 112014      MOV#B (R0)+,(CHX) ; COMMAND <CHAN-B-RESET>.
27 162312 000240      NOP ; WAIT AGAIN.
28 162314 106427 000200 1S: MTPS #PR4 ; RAISE CPU.
29 162320 026504 170010  CMP INHIB(STOP),CHX ; THIS CHANNEL INHIBITED ??
30 162324 001511      BEQ 5S ; BR IF SO.
31 162326 112013      MOVR (R0)+,(CHA) ; ELSE, POINT TO...
32 162330 112013      MOVB (R0)+,(CHA) ;...AND SET BUS-INTERFACE (CR2A ALWAYS).
33 162332 112014      MOVB (R0)+,(CHX) ; POINT TO...
34 162334 112014      MOVB (R0)+,(CHX) ;...AND SET-UP THE REST.
35 162336 105710      TSTB (R0)
36 162340 100374      BPL 2S
37 162342 012701 163666  MOV #FLT10,R1 ; XMT BUFFER POINTER => R1.
38 162346 010502      MOV STOP,R2
39 162350 062702 150002  ADD #BUFP1,R2 ; RCV BUFFER POINTER => R2.
40 162354 010246      MOV R2,-(SP) ; SAVE A COPY OF R2.
41 162356 012700 000020  MOV #16,,R0
42 162362 005022      3S: CLR (R2)+ ; CLEAR IT...
43 162364 077002      SOB R0,3S
44 162366 011602      MOV (SP),R2 ;...AND RESET POINTER.
45 162370 005065 170006  CLR STATS(STOP) ; CLEAR SAVED STATUS...
46 162374 005065 170002  CLR KAT(STUP) ;...AND KEEP-ALIVE-TIMER.
47 162400 012737 163522 000070  MOV #NEC.I,NECV ; SET VECTOR...
48 162406 012737 000200 000072  MOV #PR4,NECV+2
49 162414 106427 000000  MTPS #PRO ;...AND LOWER CPU.
50 162420 012700 000001  MOV #1,,R0 ; INIT BYTE COUNTS (RCV=0, XMT=1)...
51 162424 112164 000002  MOVB (R1)+,XDB(CHX) ;...XMIT 1ST BYTE...
52 162430 112714 000300  MOVB #<3*BIT6>,(CHX) ;...AND COMMAND <RESET-IDLE/CRC-LATCH>.
53 162434 026527 170004 162150  CMP MODE(STOP),#S.SET ; SYNC MODE ??
54 162442 001411      BEQ 41S ; BR IF SO.
55
56 162444 020027 011022 4S: CMP R0,#<18.*#IT8>|18. ; ASYNC -- 18 BYTES TRANSFERRED ??
57 162450 001423      BEQ 43S ; BR IF SO.

```

```

58 162452 005365 170002          DEC   KAT($STOP)
59 162456 001372                 BNE   4S      ; ELSE, HANG-AROUND.
60 162460 052715 000004          BIS   #BIT2,$(SEF)  ;; F2 = ASYNC MODE, DATA XFER INCOMPLETE
61 162464 000415                 BR    43$    
62
63 162466 005765 170006          41$:  TST   STATS($STOP) ; SYNC -- <EOF> RECEIVED ??
64 162472 100405                 BMI   42S    ; BP IF SO.
65 162474 005365 170002          DEC   KAT($STOP)
66 162500 001372                 BNE   41S    ; ELSR, SAME-OLE-CRAP.
67 162502 052715 000010          BIS   #BIT3,$(SEF)  ;; F3 = SYNC MODE, EOF-SDLC NOT RECEIVED
68
69 162506 020027 012022          42$:  CMP   R0,#<20.*BIT8!18.> ; BYTE COUNTS RIGHT ??
70 162512 001402                 BEQ   43S    ; PROCEED IF SO.
71 162514 052715 000020          BIS   #BIT4,$(SEF)  ;; F4 = SYNC MODE, DATA XFER INCOMPLETE
72
73 162520 010065 170002          43$:  MOV   R0,KAT($STOP) ; *** DEBUG SAVE FINAL BYTE COUNTS.
74 162524 012602                 MOV   (SP)+,R2  ; GET RCV'D...
75 162526 012701 163666          MOV   #FLT10,R1  ;...AND XMT'D POINTERS.
76 162532 012700 000022          MOV   #18,,R0
77 162536 122221                 44$:  CMPB  (R2)+,(R1)+ ; RCV'D SAME AS THAT XMT'D ??
78 162540 001402                 BEQ   45S    ; BR IF SO.
79 162542 052715 000040          BIS   #BIT5,$(SEF)  ;; F5 = SYNC/ASYNC MODES, RECEIVED DATA INCORRECT
80 162546 077005                 45$:  SUB   R0,44$    
81
82 162550 012700 162200          5$:   MOV   #C.CLR,R0  ; DONE, CLEAN-UP TABLE POINTER => R0.
83 162554 112014                 6$:   MOVB  (R0)+,(CHX) ; POINT...
84 162556 112014                 MOVB  (R0)+,(CHX) ;...AND CLEAR (RESET) AS NECESSARY.
85 162560 105710                 TSTB  (R0)
86 162562 100374                 BPL   6S
87 162564 020403                 CMP   CHX,CHA  ; BOTH CHANNELS DONE ??
88 162566 001405                 BEQ   7$      ; BR IF SO.
89 162570 010304                 MOV   CHA,CHX  ; ELSE, POINT TO CHAN A...
90 162572 016500 170004          MOV   MODE($STOP),R0  ;...GET SET-UP POINTER => R0...
91 162576 005720                 TST   (R0)+  ;...BUMP PAST THE CHAN-RESET...
92 162600 000645                 BEQ   1$      ;...AND GO 'ROUND ONCE.
93
94 162602 026527 170004 162150 7$:  CMP   MODE($STOP),#S.SET ; SYNC MODE DONE ??
95 162610 001217                 BNE   NFC.SYNC ; NO, DO IT NOW.
96
97 162612 026503 170010          CMP   INHIB($STOP),CHA ; CHAN A INHIBITED ??
98 162616 001002                 BNE   NEC.DMA  ; PROCEED TO DMA AND MODEM TESTS IF NOT...
99 162620 000137 163470          JMP   NECKIT ;...ELSE, QUIT HERE.

```

```

1 ; 5TH PASS USES DMA (ASSUMING THAT 8016 IS ALIVE AND WELL).
2 ; THE 7201 CHIP IS STILL SET-UP FOR "CHAN A SYNC" MODE.
3 ;
4 ;
5 162624 106427 000200      NEC.DMA: MTPS #PR4
6 162630 012701 162650      MOV #2S,R1 ; ADJUST CHAN A SET-UP.
7 162634 012702 000004      MOV #4,R2
8 162640 112113      1$: MOVB (R1)+,(CHA) ; POINT TO...
9 162642 112113      MOVB (R1)+,(CHA) ;...AND CHANGE...
10 162644 077203      SOB R2,1$ ;...THE FOLLOWING 4 REGISTERS.
11 162646 000404      BR 3$
12 162650 002 025      2$: .BYTE 2, 025 ; CHANGE BUS-INTERFACE TO <CHAN-A-DMA>
13 162652 003 311      .BYTE 3, 311 ; ENABLE RECRV...
14 162654 005 151      .BYTE 5, 151 ;...AND XMTR.
15 162656 001 016      .BYTE 1, 016 ; CHANGE INT CONTROL TO <RXI-ON-1ST-CHAR>
16
17 ;DEFINE THE DMA CHIP REGISTERS THAT ARE USED BELOW.
18 000070      MMR= 70 ; MASTER MODE REGISTER.
19 000056      STAT1= 56 ; CHANNEL 1 STATUS.
20 000054      CMDR= 54 ;...AND COMMAND REGISTER (FOR BOTH).
21 000046      CHA1H= 46 ; CHAIN ADDRESS REGISTERS HIGH (SEG/TAG)... .
22 000044      CHA2H= 44
23 000042      CHA1L= 42 ;...AND THE LOW (OFFSET) HALFS.
24 000040      CHA2L= 40
25 000001      TC= 1 ; TERMINAL COUNT STATUS BIT (BIT0).
26 020000      IP= 20000 ; INTERRUPT POSTED (DONE) STATUS BIT (BIT13).
27
28 162660 010500      3$: MOV $TUP,R0
29 162662 062700 150002      ADD #BUFR1,R0 ; GET BUFFER POINTER.
30 162666 012701 000020      MOV #16.,R1
31 162672 005020      35$: CLR (R0)+ ; CLEAR RECEIVING BUFFER.
32 162674 077102      SOB R1,35S
33 162676 012737 163632 000070      MOV #IXIT+2,NECV ; CHANGE THE VECTOR (INTS DISMISSED).
34 162704 012704 174400      MOV #SDMA,R4 ; SET-UP THE DMA ENGINE.
35 162710 005064 000054      CLR CMDR(R4) ; CHIP RESET.
36 162714 012764 000115 000070      MOV #115,MMR(R4) ; SET MASTER MODE <VIWAITICPINTLVENAB>.
37 162722 012764 000074 000054      MOV #40134,CMDR(R4) ; CLEAR IE'S...
38 162730 012764 000075 000054      MOV #41134,CMDR(R4) ;...BOTH CHANNELS.
39 162736 005064 000046      CLR CHA1H(R4) ; CLEAR HI CHAIN ADDRESS...
40 162742 005064 000044      CLR CHA2H(R4) ;...BOTH CHANNELS.
41 162746 012764 163004 000040      MOV #4$,CHA2L(R4) ; SET CHAN 2 (XMT)...
42 162754 012764 163024 000042      MOV #5$,CHA1L(R4) ;...AND CHAN 1 (RCV) CHAIN ADDRESSES.
43 162762 012764 000240 000054      MOV #240,CMDR(R4) ; CHAIN-LOAD CHAN 1 (RCVR)...
44 162770 060564 000002      ADD $TUP,2(R4) ;...AND ADJUST BUFR1 ADDRESS.
45 162774 012764 000241 000054      MOV #241,CMDR(R4) ; CHAIN-LOAD AND START CHAN 2 (XMTR).
46 163002 000420      BR 6$ ; CHAIN-LOAD ARA, ARB, OPK, AND CH-MODE.
47 163004 001602      4$: 1602 ; FROM -- DATA TABLE.
48 163006 000000 163666      0, FLT10 ; TO ---- 7201 XDB(LB), NO AUTO-INCR.
49 163012 000020 175707      20, 175706+1 ; OPK --- 18 BYTES.
50 163016 000022      18. ; MODE -- SFT-REQ,IP-ON-TCIEOP,SNGL,BYTE-BYTE.
51 163020 000020 001201      5$: 1602 ; FROM -- 7201 ROB(LB), NO AUTO-INCR.
52 163024 001602      20, 175702+1 ; TO ---- LOCAL BUFR1.
53 163026 000020 175703      0, BUFR1 ; OPK --- 20 BYTES (18 DATA + 2 CRC).
54 163032 000000 150002      20. ; MODE -- SAME AS ABOVE (EXCEPT NO SFT-REQ).
55 163036 000024 001201      0, 1201 ;
```

```

58 163044 112713 000300      6$:   MOVB  #<3*BIT6>,(CHA) ; COMMAND <RESET-IDLE/CRC-LATCH>...
59 163050 005000
60 163052 106400
61 163054 032764 020000 000056 7$:   CLR   R0
                                         MTPS  R0           ;...LOWER CPU.
                                         BTI   #IP,STAT1(R4) ;...AND WAIT FOR CHAN1 (RCVR) DONE.
62 163062 001002
63 163064 077005
64 163066 000404
65 163070 032764 000001 000056 8$:   BNE   8$           ; GOT <IP>, SHOULD ALSO HAVE <TC>.
66 163076 001002
67 163100
68 163100 052715 000100      9$:   BR    9$           ; BR IF SO (ALL RECEIVED).
                                         BTS   #BIT6,($EF)  ; E6 = DMA MODE, DATA XFER INCOMPLETE
69 163104 012700 163666      11$:  MOV   #FLT10,R0
70 163110 010501
71 163112 062701 150002
72 163116 012702 000022
73 163122 122021      12$:  ADD   #BUF.R1,R1
                                         MOV   #18.,R2
                                         CMPB (R0)+,(R1)+ ; CHECK DATA.
                                         BEQ   13$          ; BR IF OK.
74 163124 001402
75 163126 052715 000200      13$:  BIS   #BIT7,($EF)  ; E7 = DMA MODE RECEIVED DATA INCORRECT
76 163132 077205
77
78 163134 005037 177520      CLR   SCSRA          ; ALL DONE, CLEAR CSRA...
79 163140 005064 000054      CLR   CMDR(R4)     ;...RESET DMA CHIP AND FALL THRU.

```

W 88

```

1 ; FINALLY, TEST ALL THE MODEM CONTROL FUNCTIONS.
2 ;  

3 ;  

4     000004             CHB= $4           ; REDEFINE R4 AS CHAN B POINTER.
9 ;  

10 163144 106427 000200      NEC.MDM: MTPS #PR4          ; RAISE CPU.
11 163150 010304      MOV  CHA,CHB
12 163152 062704 000010      ADD  #10,CHB          ; SET CHAN B POINTER.
13 163156 112713 000030      1S:  MOVB #<3*BIT3>,(CHA) ; <CHAN-RESET>.
14 163162 112714 000030      MOVB #<3*BIT3>,(CHB)
15 163166 012700 000031      MOV  #100*2/5,R0    ;DELAY 100 USEC.
16 163172 077001      SOB  R0,.
17 163174 112713 000021      MOVB #<2*BIT3>!1,(CHA) ; <RESET-EXT>, POINT TO CR1...
18 163200 112714 000021      MOVB #<2*BIT3>!1,(CHB)
19 163204 112713 000005      MOVB #5,(CHA)        ;...AND SET <CAVIINT-ON-EXT>.
20 163210 112714 000005      MOVB #5,(CHB)
21 163214 012737 163640 000070      MOV  #NEC.EX,NECV ; SET THE VECTOR.
22 163222 005001      CLR  R1          ; CLEAR A SPACE FOR SRO(A)...
23 163224 005002      CLR  R2          ;...AND SRO(B).
24 ; TEST REQ-TO-SEND, CLEAR-TO-SEND, AND RCVR-READY BITS.
25 ;  

26 163226 005000      3S:  CLR  R0          ; KREP-ALIVE TIMER.
27 163230 106400      MTPS R0          ; LOWER CPU.
28 163232 112713 000005      MOVB #5,(CHA)        ; POINT TO CR5(A)... .
29 163236 112713 000002      MOVB #2,(CHA)        ;...AND SET <REQ-TO-SEND>.
30 163242 105701      4S:  TSTB R1          ; SHOULD GET SRO(A) STATUS INTERRUPT.
31 163244 001002      BNE  SS          ; BR IF SO.
32 163246 077003      SOB  R0,4$        ;  

33 163250 000406      BP   6$          ; INTERRUPT NOT RECEIVED.
34 163252 032701 000040      5S:  BIT  #BIT5,P1      ; SHOULD HAVE <CLR-TO-SEND>...
35 163256 001403      BEQ  6$          ;  

36 163260 032701 000010      BIT  #BIT3,R1      ;...AND <RCVR-READY>.
37 163264 001002      BNE  7$          ;  

38 163266 163266 052715 000400      6$:  BIS  #BIT8,(SEF)    ; E8 = STATUS WRONG OR NO INTERRUPT WITH REQ-TO-SEND SET
39 ;  

40 ; NEXT, TEST TT108/2 (IN CSRA) AND DATA-MODE BITS.
41 ;  

42 163272 005002      7S:  CLR  R2          ;  

43 163274 112737 000010 177520      MOVB #BIT3,SCSRA ; SET <TT108/2>.
44 163302 105702      8S:  TSTB R2          ; SHOULD GET SRO(B) STATUS INTERRUPT.
45 163304 001002      BNE  9$          ; BR IF SO.
46 163306 077003      SOB  R0,8$        ;  

47 163310 000406      BR   10$         ; INTERRUPT NOT RECEIVED.
48 163312 032702 000010      9S:  BIT  #BIT3,R2      ; SHOULD HAVE <DATA-MODE>...
49 163316 001403      BEQ  10$         ;  

50 163320 032702 000040      BIT  #BIT5,R2      ;...AND NOT <INCOMING-CALL>.
51 163324 001402      BEQ  11$         ;  

52 163326 163326 052715 001000      10$: BIS  #BIT9,(SEF)    ; E9 = STATUS WRONG OR NO INTERRUPT WITH TT108/2 SET
53 ;  

54 ; NEXT, TEST TERM-IN-SERV (CSRA), TT142 (CSR8), AND INCOMING-CALL BITS.
55 ;  

56 163332 005002      11S: CLP  R2          ;  

57 163334 052737 000020 177520      BIS  #BIT4,SCSRA ; SET <TERM-IN-SERV>.
58 163342 105702      12S: TSTB R2          ; SHOULD GET ANOTHER SRO(B) CHANGE.

```

```

59 163344 001002           BNE   13$  

60 163346 077003           S08   R0,12$  

61 163350 000407           BR    14$  

62 163352 032737 000001 177522 13$: BIT   #BIT0,$CSRB ; INTERRUPT NOT RECEIVED.  

63 163360 001403           BEQ   14$  

64 163362 032702 000040           BIT   #BITS5,R2 ;...AND <INCOMING-CALL>.  

65 163366 001002           BNE   15$  

66 163370           14$:  

       163370 052715 002000           BIS   #BIT10,(SEF) ;; E10 = STATUS WRONG OR NO INTERRUPT WITH TERM-IN-SERV SET  

67 ;  

68 ; FINALLY, TURN ALL THAT STUFF OFF !!!!!  

69 ;  

70 163374 042737 000010 177520 15$: BTC   #BIT3,SCSRA ; CLEAR <TT108/2>  

71 163402 012700 000031           MOV   #100*2/5,R0 ;;DELAY 100 USEC.  

       163406 077001           S08   R0,.  

72 163410 005037 177520           CLR   $CSRA ; CLEAR <TERM-IN-SERV>.  

73 163414 012700 000031           MOV   #100*2/5,R0 ;;DELAY 100 USEC.  

       163420 077001           S08   R0,.  

74 163422 112713 000005           MOVB  #5,(CHA) ; POINT TO CR5(A)...  

75 163426 112713 000000           MOVB  #0,(CHA) ;...AND CLEAR <REQ-TO-SEND>.  

76 163432 012700 000031           MOV   #100*2/5,R0 ;;DELAY 100 USEC.  

       163436 077001           S08   R0,.  

77 163440 032701 000050           BIT   #BITS1BIT3,R1 ; <CS1RR> IN SRO(A) SHOULD BE CLEAR...  

78 163444 001003           BNE   16$  

79 163446 032702 000050           BIT   #BITS1BIT3,R2 ;...AND <IC1DM> IN SRO(B) AS WELL.  

80 163452 001402           BEQ   17$  

81 163454           16$:  

       163454 052715 004000           BIS   #BIT11,(SEF) ;; E11 = STATUS WRONG WITH EVERYTHING OFF  

82 ;  

83 163460 112713 000030           17$: MOVB  #<3*BIT3>,(CHA) ; ALL DONE, RESET THE CHIP.  

84 163464 112714 000030           MOVB  #<3*BIT3>,(CHB)  

85 163470           NECXIT:  

       163470 012637 000072           MOV   (SP)+,NECV+2 ;; RESTORE NECV.  

       163474 012637 000070           MOV   (SP)+,NECV  

       163500 012637 000106           MOV   (SP)+,PEVNT+2 ;; RESTORE PEVNT.  

       163504 012637 000104           MOV   (SP)+,PEVNT  

       163510 011500           MOV   (SEF),R0 ;; ERROR BITS => R0<11:00>...  

       163512 001402           BEQ   30006$ ;;...AND SKIP IF NONE.  

       163514 052700 070000           BIS   #<7*BIT12>,R0 ;; ELSE, ADD TEST NUM => R0<15:12>...  

       163520 000207           30006$: RETURN ;;...AND RETURN (N2).

```

40

```

1 ; SYNC/ASYNC DATA TEST INTERRUPT HANDLER.
2 ; BYTE COUNTS ARE MAINTAINED IN R0 (LO=XMT, HI=RCV).
3 ;
4
5 163522 112763 000002 000010 NEC.I: MOVB #2,10(CHA) ; POINT TO SR2B...
6 163530 116346 000004 MOVB STAT+10(CHA),-(SP) ;...AND PUSH VECT/COND.
7 163534 006016 ROR (SP) ; BIT 0 => "C"...
8 163536 103420 BCS ISEXT ;...AND BR IF EXT/SPECIAL (1, 3, 5, OR 7)
9 163540 006016 ROR (SP) ; BIT 1 => "C"...
10 163542 103411 BCS IRX ;...AND BR IF RCVR (2 OR 6)...  

11 ;....ELSE FALL THRU (0 OR 4).
12
13 163544 105711 ITX: TSTB (R1) ; XMIT -- ANYTHING LEFT ??
14 163546 001404 BEQ 1$ ; BR IF NOT.
15 163550 112164 000002 MOVB (R1)+,XDB(CHX) ; SEND NEXT BYTE...
16 163554 005200 INC R0 ;...AND COUNT IT.
17 163556 000424 BR IXIT
18 163560 112714 000050 1$: MOVB #<5*BIT3>,(CHX) ; XMIT DONE, COMMAND <RESET-TXIP>...
19 163564 000421 BR IXIT ;...AND RETURN
20
21 163566 116422 177776 IRX: MOVB RDB(CHX),(R2)+ ; RCVR -- STUFF THE CHAR...
22 163572 062700 000400 ADD #<1*BIT8>,R0 ;...AND COUNT IT.
23 163576 000414 BR IXIT
24
25 163600 116465 177774 170006 ISEXT: MOVB STAT(CHX),STATS(STOP) ; EXTERNAL/SPECIAL, GET SRO...
26 163606 112714 000001 MOVB #1,(CHX)
27 163612 116465 177774 170007 MOVB STAT(CHX),STATS+1(STOP) ;...AND SR1.
28 163620 112714 000060 1$: MOVB #<6*BIT3>,(CHX) ; COMMAND <RESET-ERROR>...
29 163624 112714 000020 MOVB #<2*BIT3>,(CHX) ;...AND <RESET-EXTERNAL>
30
31 163630 005726 IXIT: TST (SP)+ ; FIX STACK.
32 163632 112713 000070 MOVB #<7*BIT3>,(CHA) ; SIGNAL <END-OF-INTERRUPT>.  

33 163636 000002 RTI ;...AND RETURN.
34
35 ; MODEM CONTROL TEST INTERRUPT HANDLER.
36
37 163640 112713 000020 NEC.EX: MOVB #<2*BIT3>,(CHA) ; COMMAND <RESET-EXTERNAL>.
38 163644 112714 000020 MOVB #<2*BIT3>,(CHB) ; DITTO.
39 163650 116301 177774 MOVB STAT(CHA),R1 ; GET SRO(A)...
40 163654 116402 177774 MOVB STAT(CHB),R2 ;...AND SRO(B).
41 163660 112713 000070 MOVB #<7*BIT3>,(CHA) ; SIGNAL <END-OF-INTERRUPT>.
42 163664 000002 RTI
43
44 ; FLOATING 1 AND 0 DATA TABLE.
45
46 163666 015 012 FLT10: .BYTE 15, 12 ; SYNC PATR.
47 163670 001 002 004 FLT1: .BYTE 001, 002, 004, 010, 020, 040, 100, 200 ; FLOATING 1.
48 163700 376 375 373 FLT0: .BYTE 376, 375, 373, 367, 357, 337, 277, 177 ; FLOATING 0.
49 163710 000000 .WORD 0 ; TERMINATOR.
50
51 .DSABL AMA

```

```
1           .SBTTL LBTEST - LOOP BACK TESTS
2
3           .ENABLE GBL,LC
4
5           ; Module name: LOOP BACK TEST - LBTFST
6
7           ; System: KXT11-CA Native Firmware
8
9
10          ;
11          ;
12          ; Functional Description:
13
14          ; This module controls the execution of the loop back tests.
15          ; The tests are:
16
17          ;     SLU1
18          ;     SLU2A
19          ;     SLU2B
20          ;     PIO
21          ;     Power Up Tests
22
23          ;
24          ;
25          ; Input Parameters:
26          ;
27          ;     None
28
29          ; Output Parameters:
30          ;
31          ;     None
32
33          ;
34          ; Routines Used:
35
36          ;     1) Execute Test (EXTEST)
37          ;     2) Power Up Tests (PUTEST)
38
39
```

```
1          LBTEST::          ;* PROCEDURE CONTINUOUS SELF TEST
2 163712          LB100:          ;* * DO UNTIL POWER DOWN
3          MOV    #CSTMD,R0      ;* * * SET LEDs TO LOOP BACK TEST DISPLAY
4 163712          CALL   SETLED
5          LB100:          ;* * * TEST SLU1 LOOP BACK
6 163712 012700 000006      MOV    #6,R2
7 163716 004767 003256      CALL   EXTEST
8          LB100:          ;* * * TEST SLU2 CHANNEL A
9 163722 012702 000006      MOV    #7,R2
10 163726 004767 006636     CALL   EXTEST
11          LB100:          ;* * * TEST SLU2 CHANNEL B
12 163732 012701 000001      MOV    #8, R1
13 163736 012702 000007      MOV    #7, R2
14 163742 004767 006622     CALL   EXTEST
15          LB100:          ;* * * TEST PIO
16 163746 012701 000002      MOV    #9, R1
17 163752 012702 000007      MOV    #7, R2
18 163756 004767 006606     CALL   EXTEST
19          LB100:          ;* * * PERFORM POWER UP SELF TESTS
20 163762 012702 000010      MOV    #10, R2
21 163766 004767 006576     CALL   EXTEST
22          LB100:          ;* * END UNTIL
23 163772 004767 006326     CALL   PUTEAT
24          LB100:          ;* END CONTINUOUS SELF TEST
25 163776 000745          BR    LB100
26
```

1 .SBTTL RESTRT - RESTART HANDLER
2 .ENABLE LC,GBL
3 . = 164000
4
5 ;
6 ; Module name: RESTRT - RESTART
7 ;
8 ; System: KXT11-CA Native Firmware
9 ;
10 ;
11 ;
12 ;
13 ;
14 ;
15 ; Functional Description:
16 ;
17 ;This module handles the restart interrupt at location 173004.
18 ;
19 ;Input Parameters:
20 ;
21 ; ; NONE
22 ;
23 ;Output Parameters
24 ;
25 ; ; NONE
26 ;
27 ;Data Structures Used
28 ;
29 ;Routines Used
30 ;
31 ; 1. SODTM (jump to)
32 ;
33 ; 2. QC0MIN (jump to)
34 ;
35 ; 3. TRAP4 (jump to)
36 ;
37 ; 4. TRAP10 (jump to)
38 ;
39 ; 5. TRAP24 (jump to)

```

1                               ;* PROCEDURE RESTART HANDLER
2 164000                      RESTRT::                         ;** * IF BREAK CAUSED RESTART
3
4 164000 132737 000020 177522   BITB  #BIT4,@#KL.CSB
5 164006 001437                BEQ   RE400
6 164010 132737 000040 177522   BITB  #BIT5,@#KL.CSB
7 164016 001433                BEQ   RE400
8 164020 132737 000100 177522   BITB  #BIT6,@#KL.CSP
9 164026 001427                BEQ   RE400
10
11
12 164030 042737 100200 177530   BIC   #<KX$DRT 1 KXSNXH>,@#KW.CSD
13
14 164036 005737 177562        TST   @#RAUF1A
15
16 164042 032737 020000 175002   BIT   #KX$BDF,@#KW.STA
17 164050 001015                BNE   RE250
18
19
20 164052 032737 000100 175002   BIT   #KX$SOF,@#KW.STA
21 164060 001005                BNE   RF200
22
23
24 164062 052737 020000 175002   BIS   #KX$BDF,@#KW.STA
25
26 164070 000167 000532        JMP   SODTM ;(NON STRUCTURED EXIT)
27
28 164074                      RE200:                         ;** * THEN
29
30 164074 062706 000004        ADD   #4,SP
31
32 164100 000167 000570        JMP   BRKENT ;(NON STRUCTURED EXIT)
33
34
35 164104                      RE250:                         ;** * THEN
36
37 164104 000002              RTI   ;(NON STRUCTURED EXIT)
38
39
40 164106                      RE400:                         ;** * END IF
41
42 164106 032737 000040 175002   BIT   #KX$FEF,@#KW.STA
43 164114 001401                BEQ   RE450
44
45
46 164116 000777              FAR::  BP
47
48 164120                      RE450:                         ;** * END IF
49
50
51 164120 032737 000200 177530   BIT   #KX$DRT,@#KW.CSD
52 164126 001405                BEQ   RF500
53
54 164130 042737 100200 177530   BIC   #<KX$DRT 1 KXSNXH>,@#KW.CSD
55
56 164136 000167 003316        JMP   QCOWIN ;(NON STRUCTURED EXIT)
57

```

KXTNF - KXT11CA Native Firmware MACRO V05.00 00:51 Page 28-1
 RESTRT - RESTART HANDLER

```

58 164142 032737 100000 177530 RE500: BIT #KXSNXM,@#KW.CSD
59 164150 001500 BEQ RE1040
60 ;* * * CLEAR NXM AND DPR INDICATORS
61 164152 042737 100200 177530 BIC #KX$DRT 1 KX$NXH>,@#KW.CSD
62 ;* * * IF STACK NXM TEST FLAG = 1
63 164160 032737 004000 175002 BIT #KXSSXT,@#KW.STA
64 164166 001442 BEQ RE900
65 ;* * * THEN
66 ;* * * CLEAR STACK NXM TEST FLAG
67 164170 042737 004000 175002 BTC #KXSSXT,@#KW.STA
68 ;* * * SET STACK ERROR FLAG OF STATUS REG.
69 164176 052737 000010 175002 BTS #KXSSEF,@#KW.STA
70 ;* * * CASE MAP SELECTION
71 ;* * * WHEN MAP BIT 2 = 0
72 164204 132737 000010 177522 BITB #KXSMP2,@#KW.CSB
73 164212 001005 BNE RE600
74 ;* * * * SAVE STACK POINTER VALUE AT TOR 1 - APPLICATION
75 ;* * * * DEFAULT STACK OFFSET
76 164214 010637 077676 MOV SP,@#RTOP1+ADSTKO
77 ;* * * * SET STACK POINTER TO THAT ADDRESS
78 164220 012706 077676 MOV #RTOP1+ADSTKO,SP
79 164224 000421 BR RE800
80 ;* * * * WHEN MAP BITS 0 AND 1 = 11
81 164226 132737 000002 177524 RE600: BITB #KXSMP0,@#KW.CSC
82 164234 001411 BEQ RE700
83 164236 132737 000004 177524 BITB #KXSMP1,@#KW.CSC
84 164244 001405 BEQ RE700
85 ;* * * * SAVE STACK POINTER VALUE AT TOR 3 - APPLICATION
86 ;* * * * DEFAULT STACK OFFSET
87 164246 010637 157676 MOV SP,@#RTOP3+ADSTKO
88 ;* * * * SET STACK POINTER TO THAT ADDRESS
89 164252 012706 157676 MOV #RTOP3+ADSTKO,SP
90 164256 000404 BR RE800
91 ;* * * ELSE
92 164260 RE700:
93 ;* * * * SAVE STACK POINTER VALUE AT TOR - APPLICATION
94 ;* * * * DEFAULT STACK OFFSET
95 164260 010637 137676 MOV SP,@#RTOP2+ADSTKO
96 ;* * * * SET STACK POINTER TO THAT ADDRESS
97 164264 012706 137676 MOV #RTOP2+ADSTKO,SP
98 ;* * * END CASE
99 164270 RE800:
100 ;* * * * EMULATE TRAP TO 4 (NON STRUCTURED EXIT)
101 164270 000167 002756 JMP TRAP4
102 ;* * * ELSE
103 164274 RE900:
104 ;* * * * IF NXM FLAG = 1
105 164274 032737 010000 175002 BIT #KXSNXF,@#KW.STA
106 164302 001404 BEQ RE1000
107 ;* * * THEN
108 ;* * * * SET THE CARRY FLAG OF RETURN PSW
109 164304 052766 000001 000002 BIS #1,2(SP)
110 164312 000417 BR RE1010
111 ;* * * ELSE
112 164314 RF1000:
113 ;* * * * CLEAR STACK ERROR FLAG
114 164314 042737 000010 175002 BIC #KXSSEF,@#KW.STA

```

```

115      ;* * * * * SET TEST STACK POINTER NXN FLAG = 1
116 164322 052737 004000 175002     BIS #KXSSXT,@#KW-STA
117      ;* * * * * TEST STACK BEFORE EMULATING TRAP
118 164330 005766 177776     TST -2(SP) ;(READ NEXT TWO STACK SLOTS, WILL GET RESTART INT. IF STACK BAD)
119 164334 005766 177774     TST -4(SP)
120      ;* * * * * SET TEST STACK POINTER NXN FLAG = 0
121 164340 042737 004000 175002     BIC #KXSSXT,@#KW-STA
122      ;* * * * * EMULATE TRAP TO 4 (NON STRUCTURED EXIT)
123 164346 000167 002700     JMP TRAP4
124      ;* * * * END IF
125 164352             RE1010:
126      ;* * * ENDIF
127 164352             RE1020:
128      ;* * WHEN HALT (TEST FOR HALT INSTRUCTION AT
129      ;* RETURN PC -2)
130 164352 162716 000002     RE1040: SUB #2,(SP)
131 164356 005776 000000     TST @(SP)
132 164362 001403     BEQ RE1045
133 164364 062716 000002     ADD #2,(SP)
134 164370 000412     BR RE1070
135 164372 062716 000002     RE1045: ADD #2,(SP)
136      ;* * * IF ENTER ODT ON HALT FLAG = 1
137 164376 032737 000020 175002     BIT #KX$OHL,@#KW-STA
138 164404 001402     BEQ RE1050,
139      ;* * * THEN
140      ;* * * ENTER SERIAL ODT
141 164406 000167 000214     JMP SODTM ;(NON STRUCTURED EXIT)
142      ;* * * ELSE
143 164412             RF1050:
144      ;* * * * EMULATE TRAP TO 10
145 164412 000167 002676     JMP TRAP10 ;(NON STRUCTURED EXIT)
146      ;* * * END IF
147 164416             RE1060:
148      ;* * WHEN BHALT
149 164416             RE1070:
150 164416 032737 001000 177530     BIT #KX$BHF,@#KW.CSD
151 164424 001411     BEQ RE1080
152      ;* * * IF SODT OR QODT FLAG NOT SET
153 164426 032737 000300 175002     BIT #<KX$SOF ! KXSQOF>,@#KW-STA
154 164434 001002     BNE RE1072
155      ;* * * THEN
156      ;* * * * EMULATE TRAP TO 24
157 164436 000167 002714     JMP TRAP24
158      ;* * * ELSE
159 164442             RE1072:
160      ;* * * * CLEAR BHALT INDICATOR
161 164442 042737 001000 177530     BIC #KX$BHF,@#KW.CSD
162      ;* * * END IF
163 164450             RE1075:
164      ;* * END CASE
165 164450             RE1080:
166      ;* * RETURN FROM INTERRUPT
167 164450 000002             RTI
168      ;* END RESTART HANDLER

```

```
1 .SBTTL SAVERG - SAVE REGISTERS
2 .ENABLE LC,CBL
3
4 ; Module name: SAVERG - SAVE REGISTERS
5 ; System: KXT11-CA Native Firmware
6 ;
7 ;
8 ;
9 ;
10 ; Functional Description:
11 ;
12 ; This module is used by both the serial and Q bus ODT monitors. It
13 ; saves the contents of the users CPU registers. This module exits
14 ; by jumping back to the ODT monitor which called it. It determines
15 ; which ODT monitor it is to return to by testing the state field of
16 ; the IOP status register. The following table describes the use of the
17 ; native firmware reserved RAM for this purpose.
18 ;
19 ;
20 ; NATIVE FIRMWARE RESERVED RAM USE
21 ;
22 ;
23 ;          USE           ADDRESS
24 ;-----+
25 ; BATTERY BACKUP FLAG 1      TOP OF NATIVE RAM
26 ; "   "   "   2              TOR - 2
27 ; BOOT TYPE FLAG WORD       TOR - 4
28 ; NATIVE FIRMWARE FLAG WORD TOR - 6
29 ; APPLICATION STACK POINTER HOLD TOR - 10
30 ;           "   PS SAVE    TOR - 12
31 ;           "   PC   "     TOR - 14
32 ;           "   SP   "     TOR - 16
33 ;           "   R5   "     TOR - 20
34 ;           "   R4   "     TOR - 22
35 ;           "   R3   "     TOR - 24
36 ;           "   R2   "     TOR - 26
37 ;           "   R1   "     TOR - 30
38 ;           "   R0   "     TOR - 32
39 ; ODT TOP OF WORKING STACK TOR - 34
40 ; ODT STACK GUARD WORD     TOR - 76
41 ; APPLICATION DEFAULT STACK TOR - 100
42 ;
43 ; This routine exits by jumping to return address of QODTM
44 ; or SODTM depending on which mode the KXT is in.
45 ;
46 ;Input Parameters:
47 ;
48 ; ALL CPU REGISTERS
49 ;
50 ;Output Parameters:
51 ;
52 ; STACK POINTER
53 ;
54 ;Routines Used:
55 ;
56 ; NONE
57 ;
```

1 ;NATIVE FIRMWARE RESERVED RAM DEFINITIONS
2 ; THESE ARE ADDRESS OFFSETS FROM THE TOP OF THE NATIVE RAM
3
4 000000 B8F10 ==: 0 ;BATTERY BACKUP FLAG 1 ADDRESS OFFSET
5 177776 BBF20 ==: BBF10 - 2 ; " " " 2 " "
6 177774 BOOTFO ==: BBF20 - 2 ;BOOT TYPE FLAG, SPECIFIES BOOT DEVICE AND TEST
7 177772 NFFWDO ==: BOOTFO - 2 ;NATIVE FIRMWARE FLAG WORD ADDRESS OFFSET
8
9 177770 APLPSO ==: NFFWDO - 2 ;APPLICATION PS SAVE ADDRESS OFFSET
10 177766 APLPCO ==: APLPSO - 2 ; " " PC " " " "
11 177764 APLSP0 ==: APLPCO - 2 ; " " SP " " " "
12 177762 APLR50 ==: APLSP0 - 2 ; " " R5 " " " "
13 177760 APLR40 ==: APLR50 - 2 ; " " R4 " " " "
14 177756 APLR30 ==: APLR40 - 2 ; " " R3 " " " "
15 177754 APLR20 ==: APLR30 - 2 ; " " R2 " " " "
16 177752 APLR10 ==: APLR20 - 2 ; " " R1 " " " "
17 177750 APLR00 ==: APLR10 - 2 ; " " R0 " " " "
18 177746 ODTSTK ==: APLR00 - 2 ;ODT WORKING TOP OF STACK STACK ADDRESS OFFSET
19 177702 ODTSTG ==: -76 ;ODT STACK GUARD WORD ADDRESS OFFSET
20 177700 ADSTKO ==: -100 ;APPLICATION DEFAULT TOP OF STACK OFFSET ADDRESS
21 ;
22 077776 RTOP1 ==: 77776 ;TOP OF RAM 1
23 137776 RTOP2 ==: 137776 ; " " " 2
24 157776 RTOP3 ==: 157776 ; " " " 3
25 000000 RAOT1 ==: 0 ;BOTTOM OF RAM 1
26 040000 RBOT2 ==: 40000 ; " " " 2
27 100000 RAOT3 ==: 100000 ; " " " 3
28 077762 SP1 ==: RTOP1 + APLSP0 ;INITIAL ODT STACK 1 ABSOLUTE ADDRESS
29 137762 SP2 ==: RTOP2 + APLSP0 ; " " " 2 " "
30 157762 SP3 ==: RTOP3 + APLSP0 ; " " " 3 " "
31 000014 SPOFF ==: -APLSP0 ;OFFSET OF STACK POINTER FROM TOP OF RAM

```

1                                         ;* PROCEDURE SAVE REGISTERS
2 164452                                     SAVERG:::
3                                         ;** CASE MAP BITS
4                                         ;** WHEN MAP BIT 2 = 0
5 164452 032737 000010 177522     BIT      #KXSMPO, @#KW.CSB
6 164460 001005                               BNE     SR100
7                                         ;** * SAVE USERS STACK POINTER AT ODT STACK 1
8 164462 010637 077762     MOV     SP, @#SP1
9                                         ;** * SET STACK POINTER = ODT STACK 1
10 164466 012706 077762    MOV     #SP1, SP
11 164472 000421                               BR     SR300
12                                         ;** * WHEN MAP BITS 0 AND 1 = 11
13 164474 032737 000002 177522   SR100: BIT      #KXSMPO, @#KW.CSB
14 164502 001411                               BFQ     SR200
15 164504 032737 000004 177522   BIT      #KXSMPO, @#KW.CSB
16 164512 001405                               BFQ     SR200
17                                         ;** * SAVE USERS STACK POINTER AT ODT STACK 3
18 164514 010637 157762     MOV     SP, @#SP3
19                                         ;** * SET STACK POINTER TO ODT STACK 3
20 164520 012706 157762     MOV     #SP3, SP
21 164524 000404                               BR     SR300
22                                         ;** ELSE
23 164526                               SR200: :
24                                         ;** * SAVE USERS STACK POINTER AT ODT STACK 2
25 164526 010637 137762     MOV     SP, @#SP2
26                                         ;** * SET STACK POINTER TO ODT STACK 2
27 164532 012706 137762     MOV     #SP2, SP
28                                         ;** END CASE
29 164536                               SR300: :
30                                         ;** * SAVE USER REGISTERS 5 THROUGH 0
31 164536 010546     MOV     R5, -(SP)
32 164540 010446     MOV     R4, -(SP)
33 164542 010346     MOV     R3, -(SP)
34 164544 010246     MOV     R2, -(SP)
35 164546 010146     MOV     R1, -(SP)
36 164550 010046     MOV     R0, -(SP)
37                                         ;** * SET R5 = TOP OF RAM ADDRESS
38 164552 010605     MOV     SP, R5
39 164554 162705 177750     SUB     #APLRO0, R5
40                                         ;** * SAVE USER PC AND PS
41 164560 016500 177764     MOV     APLSPO(R5), R0
42 164564 012065 177766     MOV     (R0)+, APLPCO(R5)
43 164570 012065 177770     MOV     (R0)+, APLPSO(R5)
44                                         ;** * REMOVE USER PS AND PC FROM USER STACK
45 164574 010065 177764     MOV     R0, APLSPO(R5)
46                                         ;** * WRITE GUARD WORD = 123456
47 164600 012765 123456 177702     MOV     #123456, ODTSTG(R5)
48                                         ;** * IF STATUS REGISTER Q BUS ODT MODE FLAG SET
49 164606 032767 000200 010166     BIT      #KXSQOF, KW.STA
50 164614 001402                               BEQ     SR400
51                                         ;** THEN
52                                         ;** * RETURN TO Q BUS ODT MONITOR (NON STRUCTURED JUMP TO Q
53 164616 000167 004622     JMP     QSRET
54                                         ;** ELSE
55 164622                               SR400: :
56                                         ;** * RETURN TO SERIAL ODT MONITOR (NON STRUCTURED JUMP TO
57 164622 000167 000026     JMP     SSRFT

```

KXTNF - K111CA Native Firmware MACRO v05.00 00:51 Page 31-1
SAVERC - SAVE DECISTEGS

```
;* END IF  
;* END SAVE REGISTERS
```

58
59

50

1 .SBTTL SODTM - SERIAL ODT MONITOR
2
3 .ENABLE LC,GBL
4 ;
5 ; Module name: SODTM - SERIAL ODT MONITOR
6 ;
7 ; System: KXT11-CA Native Firmware
8 ;
9 ;
10 ;
11 ;
12 ;
13 ;
14 ; Functional Description:
15 ;
16 ; This is the main controlling module for the serial ODT mode.
17 ; It supervises the ODT function and on completion returns
18 ; control back to the application software.
19 ; There is a secondary entry point to this module. This
20 ; entry point is used by the RESTART module when it determines
21 ; a BREAK was received on the console serial line and the IOP
22 ; was executing serial ODT. This entry causes the ODT to
23 ; start with a fresh command prompt, without effecting the
24 ; contents of the application register storage area.
25 ;
26 ; The valid ODT commands are as follows:
27 ;
28 ; kn/ - Open register n (n = 0 through 7 and S)
29 ; n/ - Open location n
30 ; / - Re-open previous location or register
31 ; m,n/ - Examine locations m through n, leaving location n open
32 ; Rm,n/ - Examine registers m through n, leaving location n open
33 ; n - Deposit n in open register or location
34 ; <LF> - Close last location or register, open next +
35 ; ^ - Close last location or register, open next -
36 ; <CR> - Close location or register
37 ; P - Proceed with execution of application
38 ; Gn - execute application at location n
39 ;
40 ; NOTE: A range examine can be aborted by pressing BREAK key.
41 ;
42 ; Input Parameters:
43 ;
44 ; NONE
45 ;
46 ; Output Parameters:
47 ;
48 ; NONE
49 ;
50 ; Routines Used:
51 ;
52 ; SAVREG
53 ; SETLED
54 ; ADDREG
55 ; EXAMIN
56 ; DISPST
57 ; DATCOM

KXTNF - KXT11CA Native Firmware MACRO V05.00 00:51 Page 32-1
SODTM - SERIAL ODT MONITOR

58 ; DEPOS
59 ; GOPROC
60

52

KXTNF - KXT11CA Native Firmware MACRO V05.00 00:51 Page 33
SODTM - SERIAL ODT MONITOR

1 ;REGISTER 3 IS USED AS THE FLAG REGISTER FOR ALL ODT ROUTINES
2 ;THE FOLLOWING ARE THE BIT DEFINITIONS FOR THIS REGISTER
3
4 000001 DCDFGB ==: 1 ;DATA/COMMAND ROUTINE - DONE FLAG
5 000002 DARFGB ==: 2 ;DATA/COMMAND ROUTINE - DATA RECV'D WITH COMM. FLAG
6 000004 APDFGB ==: 4 ;ADDRESS/REGISTER ROUTINE - DONE FLAG
7 000010 FCHFGB ==: 10 ;ADDRESS/REGISTER ROUTINE - FIRST CHAR. RECV'D FLAG
8 000020 CNDFGB ==: 20 ;SERIAL ODT MONITOR ROUTINE - CONDITION FLAG
9 000040 XOFFGB ==: 40 ;GENERAL - XOFF FLAG
10 000200 REGFGB ==: 200 ;GENERAL - REGISTER FLAG
11 100000 ERFFGB ==: 100000 ;GENERAL - ERROR FLAG
12

CC

```

1                               ;* PROCEDURE SERIAL ODT MONITOR
2 164626          SODTM::           ;* * DISABLE DUAL PORT RAM
3
4 164626 042737 000100 177530    BIC   #KXSDEN,@#KW.CSD
5                                         ;* * DISABLE BREAKS
6                                         ;* * DISABLE NXM TRAP TO 4
7                                         ;* * SET SERIAL ODT FLAG
8 164634 052737 030100 175002    BTS   #<KXSBDF | KXSNXF | KXSSOF,@#KW.STA
9                                         ;* * CLEAR Q BUS ODT FLAG
10 164642 042737 000200 175002   BIC   #KXSQOF,@#KW.STA
11                                         ;* * SET SERIAL ODT FLAG
12                                         ;* * SAVE REGISTERS (JUMP TO AND BACK FROM
13                                         ;*           SAVE REGISTERS MODULE)
14 164650 000167 177576        JMP   SAVERG
15 164654          SSRET::          ;* * SET LEDS TO SERIAL ODT MODE DISPLAY
16
17 164654 012700 000004        MOV   #SODTMD,R0
18 164660 004767 002314        CALL  SETLED
19
20 164664 105737 177562        TSTB  @#RBUFIN
21
22 164670 004767 001762        CALL  ABAUD
23
24 164674          BRKFNT::          ;* * SET BAUD RATE
25                                         ;* * CLEAR RECEIVE BUFFER
26 164674 012700 177750        MOV   #ODDTSTK+2,R0
27 164700 060500          ADD   R5,R0
28 164702 010006          MOV   R0,SP
29
30 164704 042737 020000 175002   BIC   #KXSBDF,@#KW.STA
31                                         ;* * ENABLE BREAK
32 164712 005003          CLR   R3
33                                         ;* * CLEAR ALL ODT FLAG BITS
34 164714 005002          CLR   R2
35                                         ;* * CLEAR TARGET BUFFER
36 164716 052737 010000 175002    BTS   #KXSNXF,@#KW.STA
37                                         ;* * DISABLE TRAP TO 4
38 164724 004767 000560          CALL  PCRLF ;(LOCAL CALL)
39                                         ;* * PRINT CR LF
40 164730 016501 177766          MOV   APLPCO(R5),R1
41 164734 004767 001572          CALL  DISPWD
42                                         ;* * DISPLAY PC VALUE
43 164740 042737 020000 175002    BJC   #KXSBDF,@#KW.STA
44                                         ;* * CLEAR FIRST CHARACTER FLAG
45 164746          S0100:          ;* * DO UNTIL PROCEED COMMAND
46                                         ;* * OR GO COMMAND
47 164746          S0200:          ;* * DISPLAY ODT PROMPT
48
49 164746 012701 166266          MOV   #PROMPT,R1
50 164752 004767 001370          CALL  DISPST
51                                         ;* * * CLEAR FIRST CHARACTER FLAG
52 164756 042703 000010          BIC   #FCHFCB,R3
53                                         ;* * * GET ADDRESS OR REGISTER NUMBER
54 164762 004767 000534          CALL  ADDREG
55                                         ;* * * IF ERROR FLAG = 0
56 164766 005703          TST   R3
57 164770 100002          BPL   S0300

```

KXTNF - KXT11CA Native Firmware MACRO V05.00 00:51 Page 34-1
SODTM - SERIAL ODT MONITOR

```

58 164772 000167 000414      JMP    S01850
59 164776 000106 000057      S0300:   ;* * * THEN
60                                         ;* * * * IF ADDRESS OR REGISTER NO. ENTERED
61
62 164776 032703 000002      BIT    #DARFCGB,R3
63 165002 001401 000002      BEQ    S0400   ;* * * THEN
64                                         ;* * * * SAVE FIRST TARGET
65
66 165004 010402 000002      MOV    R4,P2   ;* * * END IF
67
68 165006 000106 000054      S0400:   ;* * * CASE COMMAND FOLLOWING ADDRESS OR REGISTER NUMBER
69                                         ;* * * * WHEN ","
70
71 165006 122700 000054      CMPR   #"/,R0
72 165012 001023 000002      BNE    S0900   ;* * * * GET ANOTHER ADDRESS OR REGISTER NUMBER
73
74 165014 004767 000502      CALL   ADDREG  ;* * * * IF VALID ADDRESS OR REGISTER NUMBER
75
76 165020 005703 000002      TST    R3
77 165022 100414 000002      BMI    S0700   ;* * * * AND COMMAND FOLLOWING ADDRESS OR
78                                         ;* * * REGISTER NUMBER IS "/"
79
80 165024 122700 000057      CMPB   #"/,R0
81 165030 001011 000002      BNE    S0700   ;* * * * AND ADDRESS OR REGISTER NUMBER GREATER
82                                         ;* * * THEN FIRST TARGET
83
84 165032 020402 000002      CMP    R4,R2
85 165034 101407 000002      BLOS   S0700   ;* * * * THEN
86                                         ;* * * * DISPLAY DATA OF RANGE
87
88                                         ;R2=TARGET 1
89                                         ;R4=TARGET 2
90 165036 012700 000002      MOV    #RMGFCGB,R0
91 165042 004767 000674      CALL   EXAMIN  ;(PASS TARGET 1)
92                                         ;(PASS TARGET 2)
93 165046 052703 000020      BIS    #CNDFCGB,R3
94 165052 000402 000002      BR    S0800   ;* * * * SET CONDITION FLAG = OPEN
95
96 165054 000445 000002      S0700:   ;* * * * ELSE
97                                         ;* * * * SET ERROR FLAG
98 165054 052703 100000      BIS    #ERRFCGB,R3
99                                         ;* * * * END IF
100 165060 000445 000002     S0800:   BP    S01450  ;* * * * WHEN "/*"
101
102 165062 122700 000057     S0900:   ;* * * * THEN
103 165062 001006 000002     CMPR   #"/,R0
104 165066 001006 000002     BNE    S01000  ;* * * * DISPLAY DATA
105                                         ;(PASS SERIAL ODT FLAG = 1, RANGE FLAG = 0)
106 165070 005000 000002     CLR    R0
107                                         ;(PASS TARGET 1)
108 165072 004767 000644     CALL   EXAMIN  ;* * * * SET CONDITION FLAG = OPEN
109
110 165076 052703 000020     BIS    #CNDFCGB,R3
111 165102 000434 000002     BR    S01450  ;* * * * WHEN "G"
112
113 165104 122700 000107     S01000: ;* * * * THEN
114 165104 000107 000002     CMPB   #"/,R0

```

115 165110 001014 BNE S01300 ;* * * * AND TARGET IS AN ADDRESS
116
117 165112 105703 TSTB R3
118 165114 100412 BMI S01300 ;* * * * SET LEDS TO NON NATIVE CODE DISPLAY
119
120 165116 012700 000000 MOV #NNCED,R0
121 165122 004767 002052 CALL SETLED ;* * * * EXECUTE AT ADDRESS ENTERED (WILL NOT
122
123
124 165126 010400 MOV R4,R0
125 165130 004767 001636 CALL GO ;* * * * SET ERROR FLAG = 1
126
127 165134 052703 100000 BIS #ERRFCB,R3
128 165140 000415 B0 S01450 ;* * * * WHEN "P"
129
130 165142 S01300: CMPR #'P,R0
131 165142 122700 000120 BNE S01400 ;* * * * AND NO DATA PRECEDED COMMAND
132 165146 001010
133
134 165150 032703 000002 BIT #DARFCB,R3
135 165154 001005 BNE S01400 ;* * * * EXECUTE PROCEED (WILL NOT RETURN IF
136
137
138 165156 004767 001626 CALL PROC
139
140 165162 052703 100000 BIS #ERRFCB,R3
141 165166 000402 BR S01450 ;* * * * SET ERROR FLAG = 1
142
143 165170 S01400: ;* * * * ELSE (INVALID COMMAND)
144
145 165170 052703 100000 BIS #ERRFCB,R3 ;* * * * SET ERROR FLAG
146
147 165174 S01450: ;* * * * END CASE
148
149 165174 S01470: ;* * * * DO WHILE CONDITION FLAG = OPEN
150 165174 032703 000020 BIT #CNDFCB,R3
151 165200 001504 BEQ S01800 ;* * * * AND ERROR FLAG = 0
152
153 165202 005703 TST R3
154 165204 100502 BMI S01800 ;* * * * PRINT SPACE
155
156 165206 012701 166272 MOV #SPACE,R1
157 165212 004767 001130 CALL DISPST ;* * * * GET INPUT DATA
158
159 165216 004767 001162 CALL DATCOM ;* * * * IF VALID DATA AND COMMAND
160
161 165222 005703 TST R3
162 165224 100467 BMI S01600 ;* * * * THEN
163
164
165 165226 032703 000002 BIT #DARFCB,R3
166 165232 001402 BEQ S01500 ;* * * * IF DATA PROCEEDED COMMAND
167
168
169
170
171 165234 004767 001234 CALL DEPOS ;* * * * THEN
;* * * * * DEPOSIT DATA INTO ADDRESS OR REGISTER
; (R2=ADDR.)
; (R4=DATA)

96

```

172
173 165240           SO1500:          ;* * * * * END IF
174
175
176 165240 122700 000015      CMPR  #CR,R0
177 165244 001003          BNE   SO1550  ;* * * * * CASE COMMAND
178
179 165246 042703 000020      BIC   #CNDFGB,R3
180 165252 000453          BP    SO1570  ;* * * * * SET CONDITION FLAG = CLOSED
181
182 165254           SO1550:          ;* * * * * WHEN LF
183 165254 122700 000012      CMPB  #LF,R0
184 165260 001023          BNE   SO1560  ;* * * * * PRINT CR LF
185
186 165262 004767 000222      CALL  PCRLF ;(LOCAL CALL)
187
188 165266 005202          INC   R2   ;* * * * * INCREMENT TARGET
189
190 165270 105703          TSTR  R3   ;* * * * * IF TYPE IS REGISTER
191 165272 100007          BPL   SO1555
192
193
194 165274 022702 000011      CMP   #11,R2
195 165300 101001          BHI   SO1552' ;* * * * * THEN
196 165302 005002          CLR   R2   ;* * * * * SET TARGET MOD 10
197 165304           SO1552:          ;* * * * * DISPLAY REGISTER NUMBER
198
199 165304 004767 000140      CALL  BR   ;(LOCAL CALL)
200 165310 000404          SO1557  ;* * * * * ELSE
201
202 165312           SO1555:          ;* * * * * INCREMENT TARGET AGAIN
203
204 165312 005202          INC   R2   ;* * * * * DISPLAY TARGET
205
206 165314 010201          MOV   R2,P1
207 165316 004767 001210      CALL  DISPWD ;* * * * * END IF
208
209 165322           SO1557:          ;* * * * * EXAMINE
210
211 165322 004767 000142      CALL  BR   ;(LOCAL CALL)
212 165326 000425          SO1570  ;* * * * * WHEN ^
213
214 165330           SO1560:          ;* * * * * WHEN ^
215 165330 122700 000136      CMPB  #^,R0
216 165334 001022          BNE   SO1570  ;* * * * * PRINT CR LF
217
218 165336 004767 000146      CALL  PCRLF ;(LOCAL CALL)
219
220 165342 105703          TSTR  R3   ;* * * * * IF TYPE IS REGISTER
221 165344 100007          BPL   SO1567
222
223
224 165346 005302          DEC   R2   ;* * * * * THEN
225
226 165350 100002          BPL   SO1565 ;* * * * * DECREMENT TARGET
227 165352 012702 000010      MOV   #10,R2 ;* * * * * SET REGISTER NUMBER MOD 10
228 165356           SO1565:          ;* * * * *

```

57

```

229
230 165356 004767 000066          CALL    DISPRG ;(LOCAL CALL)      ;* * * * * DISPLAY REGISTER NUMBER
231 165362 000405          BP      SD1569
232
233 165364          SD1567:          ;* * * * * ELSE
234
235 165364 005302          DEC     R2      ;* * * * * DECREMENT TARGET BY 2
236 165366 005302          DEC     R2
237
238 165370 010201          MOV     R2,R1
239 165372 004767 001134          CALL    DTSPWD      ;* * * * * DISPLAY TARGET
240
241 165376          SD1569:          ;* * * * * END IF
242
243 165376 004767 000066          CALL    EXMDIS ;(LOCAL CALL)      ;* * * * * EXAMINE
244
245 165402          SD1570:          BP      SD1700      ;* * * * * END CASE
246 165402 000402
247
248 165404          SD1600:          ;* * * * * ELSE
249
250 165404 052703 100000          BIS     #ERRFCB,R3      ;* * * * * SET ERROR FLAG
251
252 165410          SD1700:          ;* * * * * END IF
253
254 165410 000671          BR     SD1470      ;* * * * * END WHILE
255 165412          SD1800:          ;* * * * * END IF
256
257 165412          SD1850:          ;* * * * * IF ERROR FLAG = 1
258
259 165412 005703          TST     R3
260 165414 100011          BPL     SD1900      ;* * * THEN
261
262
263 165416 012701 166264          MOV     #ERROR,R1      ;* * * * * SEND ERROR PROMPT
264 165422 004767 000720          CALL    DISPST
265
266 165426 042703 000200          BIC     #REGFGL,R3      ;* * * * * CLEAR REGISTER FLAG
267
268
269 165432 012702 177777          MOV     #177777,R2      ;* * * * * SET TARGET2 TO MAX ADDRESS
270
271 165436 005004          CLR     R4      ;* * * * * (TO CAUSE AN ERROR IF RETURN / ENTERED)
272
273 165440          SD1900:          ;* * * * * CLEAR TARGET1
274
275
276 165440 042703 177577          BIC     #^C<REGFCB>,R3      ;* * * * * END IF
277
278 165444 000167 177276          JMP     SD100       ;* * * * * CLEAR ALL ODT FLAGS EXCEPT
279
280
281          ;* * * * * REGISTER FLAG
282          ;* * * * * END UNTIL
283
284          ;* * * * * END SERIAL ODT MONITOR

```

1 ; The following subroutines were created to conserve memory and are only used
2 ; by this module.
3
4
5 165450 DISPRG: ;* PROCEDURE DISPLAY REGISTER NUMBER
6
7 165450 010201 MOV R2,R1 ;* * CALCULATE DISPLAY POINTER
8 165452 006301 ASL #1
9 165454 006301 ASL R1
10 165456 062701 166303 ADD #REGDIS,R1
11
12 165462 004767 000660 CALL DISPST ;* * DISPLAY STRING
13
14 165466 000207 RETURN ;* END DISPLAY REGISTER NUMBER
15
16
17 165470 EXMDIS: ;* PROCEDURE EXAMINE AND DISPLAY
18
19
20 165470 012701 166301 MOV #OPENDS,R1 ;* * DISPLAY "/"
21 165474 004767 000646 CALL DISPST
22
23 165500 005000 CLR R0 ;* * EXAMINE DATA @ TARGET
24 165502 004767 000234 CALL EXAMIN
25
26 165506 000207 RETURN ;* END EXAMINE AND DISPLAY
27
28
29 165510 PCRLF: ;* PROCEDURE PRINT CRLF
30
31
32 165510 012701 166274 MOV #CRLF,R1 ;* * POINT TO STRING
33
34 165514 004767 000626 CALL DISPST ;* * DISPLAY STRING
35
36 165520 000207 RETURN ;* END PRINT CRLF
37

50

```
1           .SBTTL ADDREG - GET ADDRESS/REGISTER
2           .ENABLE LC,GBL
3
4           ;
5           ; Module name: ADDREG - GET ADDRESS/REGISTER
6           ;
7           ; System: KXT11-CA Native Firmware
8           ;
9           ;
10          ;
11          ;
12          ; Functional Description:
13          ;
14          ; This module is used by the serial ODT monitor. It receives data
15          ; from the console serial line, accumulates a word target delimited
16          ; by a valid command or an error. The valid commands are: ",", CR, LF,
17          ; "/", "G" or "P". It returns the target, the command
18          ; the target type (register or address) and the error status.
19
20          ;Input Parameters:
21
22          ;      NONE
23
24          ;Output Parameters
25
26          ;      1) R0 = COMMAND RECEIVED
27          ;      2) R4 = TARGET (ADDRESS OR REGISTER NO.)
28          ;      3) R3, BIT REGFGB - 1 = REGISTER NO., 0 = ADDRESS
29          ;      4) R3, BIT DARFGB - ADDRESS/REGISTER RECEIVED WITH COMMAND FLAG,
30          ;                     1 = YES. 0 = NO
31          ;      5) R3, BIT ERRFGB - 1 = ERROR, NOT ALTERED IF NO ERROR
32
33          ;DATA STRUCTURES USED
34
35          ;      NONE
36
37          ;Routines Used
38
39          ;      1) CHKCHR
40          ;      2) ACCUM
41
```

```

1          ;* PROCEDURE GET ADDRESS/REGISTER
2 165522   ADDREG::                                ;** * CLEAR ADDRESS/REGISTER DONE FLAG
3          ;** * CLEAR DATA RECEIVED FLAG
4          ;** * CLEAR ACCUMULATED NUMBER
5 165522  042703  000006      BIC  #<ARDFCB 1 DARFCB>,R3
6          ;** * DO UNTIL DONE FLAG SET
7 165526  005004      CLR  R4
8          ;** * DO UNTIL NON NUMBER CHARACTER RECEIVED
9 165530
10         AD100:
11 165530      AD200:
12          CALL  GETIN
13 165530  004767  006016      BCS  AD253
14          ;** * * * IF INPUT = 0 THRU 7
15 165534  103417
16          BIT   #FCHFCB,R3
17 165536  032703  000010      BNE  AD220
18 165542  001002
19          ;** * * * THEN
20          ;** * * * * CLEAR REGISTER FLAG
21 165544  042703  000200      BIC  #REGFCB,R3
22          ;** * * * * END IF
23 165550      AD220:
24          TSTB R3
25 165550  105703      BPL  AD240
26 165552  100002
27          ;** * * * * THEN
28          ;** * * * * REPLACE ACCUM VALUE WITH INPUT
29 165554  010004      MOV  R0,R4
30 165556  000402      BR   AD250
31          ;** * * * * ELSE
32 165560      AD240:
33          CALL  ACCUM
34 165560  004767  005754      BIS  #FCHFCB,R3
35          ;** * * * * ACCUMULATE DATA
36 165564
37          AD250:
38 165564  052703  000010      BIS  #FCHFCB,R3
39          ;** * * * * SET FIRST CHARACTER RECEIVED FLAG = 1
40 165570  000241
41 165572  000424      CLC  BR   AD290
42          ;** * * * * ELSE
43 165574      AD253:
44          CMPP R0,#"a"
45 165574  120027  000141      BLO  AD255
46 165600  103402
47          ;** * * * * THEN
48          ;** * * * * CONVERT TO UPPER CASE
49 165602  042700  000040      BIC  #40,R0
50          ;** * * * * END IF
51 165606      AD255:
52          CMPB #"$,R0
53 165606  122700  000123      BNE  AD280
54 165612  001013
55 165614      AD270:
56          TSTB R3
57 165614  105703

```

```

58 165616 100011          BPL   AD280
59                                         ;* * * * * AND NOT FIRST CHARACTER RECEIVED
60 165620 032703 000010      BIT   #FCHFCGB,R3
61 165624 001406            BEQ   AD280
62                                         ;* * * * * SET ACCUMULATED VALUE = 10
63 165626 012704 000010      MOV   #10,R4
64                                         ;* * * * * SET DATA RECEIVED FLAG = 1
65                                         ;* * * * * SET FIRST CHARACTER RECEIVED FLAG = 1
66 165632 052703 000012      BIS   #<DARFGB | FCHFCGB>,R3
67                                         ;* * * * * SET NON NUMERIC FLAG = 0
68 165636 000241            CLC
69 165640 000401            BR    AD290
70                                         ;* * * * * ELSE
71 165642                      AD280:
72                                         ;* * * * * SET NON NUMERIC INPUT FLAG = 1
73 165642 000261            SEC
74                                         ;* * * * * END IF
75 165644                      AD290:
76                                         ;* * * * * END IF
77                                         ;* * * * * END UNTIL
78 165644 103331            BCC   AD200
79 165646                      AD300:
80                                         ;* * * CASE COMMAND RECEIVED
81                                         ;* * * WHEN ","
82                                         ;* * * OR CR
83                                         ;* * * OR LF
84                                         ;* * * OR "/"
85                                         ;* * * OR "G"
86                                         ;* * * OR "D"
87 165646 012701 165734      MOV   #CHTAB,R1
88 165652 122100            CMPB  (R1)+,R0
89 165654 001404            BEQ   AD500
90 165656 022701 165742      CMP   #CHEND,R1
91 165662 001404            BEQ   AD600
92 165664 000772            BR    AD400
93 165666                      AD500:
94                                         ;* * * * * SET FIRST CHARACTER FLAG = 1
95                                         ;* * * * * SET DONE FLAG = 1
96 165666 052703 000014      BIS   #<FCHFCGB | ARDFGB>,R3
97 165672 000414            BR    AD800
98                                         ;* * * WHEN "R"
99 165674                      AD600:
100 165674 122700 000122     CMPB  #",R0
101 165700 001007            BNE   AD700
102                                         ;* * * AND FIRST CHARACTER RECEIVED FLAG = 0
103 165702 032703 000010      BIT   #FCHFCGB,R3
104 165706 001004            BNE   AD700
105                                         ;* * * * * SET REGISTER FLAG = 1
106                                         ;* * * * * SET FIRST CHARACTER FLAG = 1
107                                         ;* * * * * SET DATA RECEIVED FLAG = 1
108 165710 052703 000212      BIS   #<DARFGB | REGFGB | FCHFCGB>,R3
109                                         ;* * * * * CLEAR TARGET
110 165714 005004            CLR   R4
111 165716 000402            BR    AD800
112                                         ;* * * ELSE
113 165720                      AD700:
114                                         ;* * * * * SET ERROR FLAG

```

KXTNF - KXT11CA Native Firmware MACRO V05.00 00:51 Page 37-2
ADDREG - GET ADDRESS/REGISTER

```
115 ;* * * * SET DONE FLAG
116 165720 052703 100004     BIS    #<ERRFCB ! ARDFCB>,R3
117 ;* * * END CASE
118 165724          AD800:
119 ;* * END UNTIL
120 165724 032703 000004     BIT    #ARDFCB,R3
121 165730 001677          BFQ    AD100
122 ;* * RETURN
123 165732 000207          RETURN
124 ;* END GET ADDRESS/REGISTER
125
126 ;CHARACTER TABLE
127
128
129 165734 054 015 012 CHTAB: .ASCII ",<15><12>"/GP"
130 165742          CHEND:
131 .EVEN
```

1 .SBTTL EXAMIN - EXAMINE
2 .ENABLE LC, GBL
3
4 ;
5 ; Module name: EXAMIN - EXAMINE
6 ;
7 ; System: KXT11-CA Native Firmware
8 ;
9 ;
10 ;
11 ;
12 ;
13 ;
14 ; Functional Description:
15 ;
16 ; This module performs the memory or register examine function for
17 ; both the Q bus and serial ODT monitors. The output of this module
18 ; is directly to the user (the serial line or the dual port RAM).
19 ;
20 ;Input Parameters:
21 ;
22 ; R0 - FLAGS,
23 ; BIT (RNGFCB) = 1 (RANGE EXAMIN), BIT = 0 (SINGLE EXAMIN)
24 ; R2 - TARGET 1
25 ; BIT 0 - TYPE, BIT = 0 (REGISTER), BIT = 1 (ADDRESS)
26 ; R4 - TARGET 2
27 ; BIT 0 - TYPE, BIT = 0 (REGISTER), BIT = 1 (ADDRESS)
28 ;
29 ;Output Parameters
30 ;
31 ; R3, BIT (ERRFCB) - ERROR FLAG, BIT = 1 (ERROR), NO ERROR BIT NOT ALTERED
32 ;
33 ;DATA STRUCTURES
34 ;
35 ; 1) PRINT STRINGS
36 ; 2) DUAL PORT RAM REGISTER 2
37 ; 3) IOP STATUS REGISTER
38 ; 4) IOP COMMAND REGISTER
39 ;
40 ;Routines Used
41 ;
42 ; 1) DISPLAY STRING (DISPST)
43 ; 2) DISPLAY WORD (DISPWD)
44 ; 3) CHECK CHARACTER (CHKCHR)
45 ;
46 ;

```

1      000002          RNGFGB    ==:    2
2
3
4
5
6
7 165742          EXAMIN:::           ;* PROCEDURE EXAMINE
8
9 165742 032737 000100 175002      BIT    #KXSSOF,@#KW STA
10 165750 001525                   BEQ    EX1000
11
12
13 165752 105703          TSTB   R3
14 165754 100021          BPL    EX300
15
16
17 165756 022702 000011      CMP    #11,R2
18 165762 101001          BHI    EX10
19 165764 005002          CLR    R2
20 165766          EX10:             ;* * * THEN
21
22 165766 022704 000011      CMP    #11,R4
23 165772 101001          BHI    EX20
24 165774 005004          CLR    R4
25 165776          EX20:             ;* * * * SET TARGET 1 TO MOD 10
26
27
28 165776 006302          ASL    R2
29 166000 062702 177750      ADD    #APLR00,R2
30 166004 060502          ADD    R5,R2
31
32
33 166006 006304          ASL    R4
34 166010 062704 177750      ADD    #APLR00,R4
35 166014 060504          ADD    R5,R4
36 166016 000404          BR     EX400
37
38 166020          EX300:            ;* * * ELSE
39
40 166020 042702 000001      BIC    #BIT0,R2
41 166024 042704 000001      BIC    #BIT0,R4
42
43 166030          EX400:            ;* * * END IF
44
45 166030 032700 000007      BIT    #RNGFGB,R0
46 166034 001001          BNE    EX500
47
48
49 166036 010204          MOV    R2,R4
50
51 166040          EX500:            ;* * * IF NOT RANGE EXAMINE
52
53 166040          EX600:            ;* * * THEN
54
55
56
57

```

5

```

58 166040 000241      CLC
59 166042 012201      MOV     (R2)+,R1      ;* * * * IF NM
60
61 166044 000240      NOP
62 166046 103003      BCC     EX700       ;* * * * THEN
63
64
65 166050 052703 100000   BTS     #ERRFGB,R3  ;* * * * SET ERROR FLAG
66
67 166054 000502      BR      EX1300      ;* * * * EXIT EXAMINE (NON STRUCTURED EXIT)
68
69 166056
70
71 166056 004767 000450      EX700: CALL    DISPWD      ;* * * * DISPLAY DATA
72
73 166062 020204      CMP     R2,R4      ;* * * * IF TARGET 1 = TARGET 2
74 166064 101402      BLOS    EX750       ;* * * * THEN
75
76
77 166066 042700 000002      BIC     #RNGFGB,R0  ;* * * * SET RANGE FLAG = 0
78
79 166072
80
81 166072 032700 000002      EX750: BIT     #RNGFGB,R0  ;* * * * IF RANGE EXAMINE
82 166076 001427      BEQ     EX800       ;* * * * THEN
83
84
85 166100 012701 166274      MOV     #CRLF,R1  ;* * * * DISPLAY CR LF
86 166104 004767 000236      CALL   DISPST      ;* * * * IF TYPE IS REGISTER
87
88 166110 105703      TSTP    R3          ;* * * * ELSE
89 166112 100012      BPL     EX760       ;* * * * END IF
90
91
92 166114 010201      MOV     R2,R1
93 166116 160501      SUB    R5,R1
94 166120 162701 177750      SUB    #APLRO0,R1
95 166124 006301      ASL     R1
96 166126 062701 166303      ADD    #REGDIS,R1
97 166132 004767 000210      CALL   DISPST      ;* * * * DISPLAY REGISTER NUMBER
98 166136 000403      BP     EX770       ;* * * * ELSE
99
100 166140
101
102 166140 010201      EX760: MOV     R2,R1      ;* * * * DISPLAY ADDRESS
103 166142 004767 000364      CALL   DISPWD      ;* * * * END IF
104
105 166146
106
107 166146 012701 166301      EX770: MOV     #OPENDS,R1  ;* * * * DISPLAY "/"
108 166152 004767 000170      CALL   DISPST      ;* * * * END IF
109
110 166156
111
112 166156 005703      EX800: TST    R3
113 166160 100404      BMI    EX900       ;* * * * END UNTIL
114 166162 005702      TST    R2

```

```

    115 166164 001402           BEQ    EX900
    116 166166 020204           CMP    R2,R4
    117 166170 101723           BLOS   EX600
    118 166172
    119
    120 166172 162702 000002      EX900:          SUB    #2,R2      ;* * * SET TARGET BACK 2
    121
    122 166176 105703           TSTB   R3
    123 166200 100010           BPL    EX970      ;* * * IF TYPE IS REGISTER
    124
    125
    126 166202 160502           SUB    R5,R2      ;* * * THEN
    127 166204 162702 177750      SUB    #APLRO0,R2
    128 166210 006202           ASR    R2      ;* * * RESTORE TARGET 1 TO REGISTER NO.
    129
    130 166212 022702 000011      CMP    #11,R2
    131 166216 101001           BPI    EX950
    132 166220 005002           CLR    R2
    133 166222
    134
    135 166222
    136 166222 000417           EX950:          BR     EX1300  ;* * * END IF
    137
    138 166224
    139
    140 166224 010246           MOV    R2,-(SP) ;* * * SAVE TARGET
    141
    142 166226 105703           TSTB   R3
    143 166230 100004           BPL    EX1050  ;* * * IF TARGET IS REGISTER
    144
    145
    146
    147 166232 006302           ASL    R2      ;* * * THEN
    148 166234 062702 177750      ADD    #APLRO0,R2
    149 166240 060502           ADD    R5,R2      ;* * * SET TARGET =
    ; (2*TARRGET) + REGISTER 0 SAVE ADDRESS OFFSET
    150
    151 166242
    152
    153 166242 000241           CLC
    154 166244 011237 175004      MOV    (R2),#KW.SC2 ;* * * GET DATA PER TARGET AND PLACE IN DPR 2
    155
    156 166250 000240           NOP
    157 166252 103002           BCC    EX1200  ;* * * IF NXM
    158
    159
    160 166254 052703 100000      BTS    #ERRFCB,R3 ;* * * THEN
    ;* * * SET ERROR FLAG
    161
    162 166260
    163
    164 166260 012602           EX1200:          MOV    (SP)+,R2 ;* * * END IF
    165
    166 166262
    167
    168 166262 000207           EX1300:          RETURN ;* * * RESTORE TARGET
    169
    ;* * * END IF
    ;* * * RETURN
    ;* END EXAMINE
  
```

```
1 .SBTTL DISPST - DISPLAY STRING
2 .ENABLE LC,G8L
3
4 ; Module name: DISPST - DISPLAY STRING
5 ; System: KXT11-CA Native Firmware
6 ;
7 ;
8 ;
9 ;
10 ;
11 ;
12 ;
13 ;
14 ; Functional Description:
15 ;
16 ; This module is a general use utility that sends an ASCII string
17 ; out the console serial line. The display string is specified by
18 ; passing the address of the first character to the routine. The
19 ; end of the string is marked by a null character.
20
21 ;INPUT PARAMETERS:
22 ; 1) R1 - Start of string address /
23 ;OUTPUT PARAMETERS:
24 ;
25 ;NONE
26 ;
27 ;DATA STRUCTURES USED:
28 ;
29 ; 1) SLU1 CSRs
30
31
```

KXTNF - KXT11CA Native Firmware MACRO V05.00 00:51 Page 41
DISPST - DISPLAY STRING

1
2 000015 CR ==: 15
3 000012 LF ==: 12
4
5
6 166264 077 000 ERROR:: .ASCIZ "?"
7 166266 015 012 100 PROMPT:: .ASCIZ <CR><LF>"@"
8 166272 040 000 SPACE:: .ASCIZ " "
9 166274 015 012 000 CRLF:: .ASCIZ <CR><LF>
10 166277 015 000 CRDS:: .ASCIZ <CR>
11 166301 057 000 OPENDS:: .ASCIZ "/"
12 166303 122 060 000 REGDIS:: .ASCIZ "R0"
13 166306 .BLKB 1
14 166307 122 061 000 .ASCIZ "R1"
15 166312 .BLKB 1
16 166313 122 062 000 .ASCIZ "R2"
17 166316 .BLKB 1
18 166317 122 063 000 .ASCIZ "R3"
19 166322 .BLKB 1
20 166323 122 064 000 .ASCIZ "R4"
21 166326 .BLKB 1
22 166327 122 065 000 .ASCIZ "R5"
23 166332 .BLKB 1
24 166333 122 066 000 .ASCIZ "R6"
25 166336 .BLKB 1
26 166337 122 067 000 .ASCIZ "R7"
27 166342 .BLKB 1
28 166343 122 123 000 .ASCIZ "RS"
29
30 .EVEN
31

```
1                               ;* PROCEDURE DISPLAY STRING
2 166346      DISPST::          ;* * DO UNTIL NULL CHARACTER POINTED TO
3                               DS100:           ;* * * DO UNTIL TRANSMIT BUFFER EMPTY
4 166346      DS200:           ;* * * OR XOFF FLAG = 1
5                               ;* * * TEST FOR INPUT
6 166346
7
8
9 166346 010046
10 166350 004767 005240      MOV    R0,-(SP)
11 166354 012600             CALL   CHKCHR
12                               MOV    (SP)+,R0
13 166356 105737 177564      TSTB  @#XCSR1A
14 166362 100371             BPL   DS200
15 166364 032703 000040      BIT    #XUFFGB,R3
16 166370 001366             BNE   DS200
17                               ;* * * END UNTIL
18 166372 112137 177566      MOVB  (R1),@#XBUF1A
19                               ;* * * SEND CHARACTER PER POINTER
20 166376 105711             TSTB  (R1)
21 166400 001362             BNE   DS100
22                               ;* * END UNTIL
23 166402 000207             RETURN
24                               ;* * RETURN FROM CALL
                               ;* END DISPLAY STRING
```

1 .SBTTL DATCOM - GET DATA/COMMAND
2 .ENABLE LC, GBL
3
4 ;
5 ; Module name: DATCOM - GET DATA/COMMAND
6 ;
7 ; System: KXT11-CA Native Firmware
8 ;
9 ;
10 ;
11 ;
12 ;
13 ; Functional Description:
14 ;
15 ; This module is used by the serial ODT monitor. It polls SLU1
16 ; for valid data and returns when a CR, LF OR ^ are received
17 ; or on an error.
18 ;
19 ;Input Parameters:
20 ;
21 ; NONE
22 ;
23 ;Output Parameters
24 ;
25 ; 1) R0 - COMMAND RECEIVED
26 ; 2) R4 - INPUT DATA
27 ; 3) R3 - BIT ERRFGB, ERROR FLAG: 1 = ON ERROR, NOT ALTERED IF NO ERRORS
28 ; 4) R3 - BIT DARFGB, DATA RECV'D WITH COMMAND FLAG: 1 = YES, 0 = NO
29 ;
30 ;DATA STRUCTURES USED
31 ;
32 ; NONE
33 ;
34 ;Routines Used
35 ;
36 ; 1) ACCUM
37 ; 2) CHKCHR
38

71

```
1          ;* PROCEDURE GET DATA/COMMAND
2 166404      DATCOM:::                         ;* * CLEAR DATA RECEIVED FLAG
3                                     ;* * CLEAR DATA/COMMAND DONE FLAG
4
5 166404 042703 000003      BIC    #<DARFGB ! DCDFGB>,R3   ;* * CLRAR ACCUMULATED NUMBER
6                                     CLR    R4
7 166410 005004      DA100:                      ;* * DO UNTIL DONE FLAG SET
8                                     DA200:                      ;* * * * DO UNTIL NON NUMERIC INPUT
9 166412      CALL   GFTIN                     ;* * * * * GET INPUT
10 166412      DA370:                      ;* * * * * IF NUMERIC INPUT
11 166412      CALL   ACCUM                    ;* * * * * ACCUMULATE NUMBER
12 166412 004767 005134      CLC
13 166412 000241      DA370:                      ;* * * * * SET NON NUMERIC FLAG = 0
14 166416 103403      HCC    DA200
15 166416      DA370:                      ;* * * * * END IF
16 166420 004767 005114      CMPB  #15,R0
17 166420      BEQ   DA400                    ;* * * * * END UNTIL
18 166424      CALL   ACCUM
19 166424 000241      CLC
20 166426      DA370:                      ;* * * * * CASE CHARACTER INPUT
21 166426 103371      HCC    DA200
22 166426      DA370:                      ;* * * * * WHEN CR
23 166430 122700 000015      CMPB  #12,R0
24 166430      BEQ   DA400                    ;* * * * * OR LF
25 166434 001406      CMPB  #12,R0
26 166434      BEQ   DA400
27 166434 001403      CMPB  #12,R0
28 166436 122700 000012      BEQ   DA400
29 166436 001403      CMPB  #12,R0
30 166442 001003      BNE   DA500
31 166444 122700 000136      CMPB  #12,R0
32 166444 001003      BNE   DA500
33 166450 000402      DA400:                      ;* * * * * SET DONE FLAG
34 166452 052703 000001      BIS   #DCDFGB,R3
35 166452 000402      BR    DA600
36 166452 052703 000001      BIS   #DCDFGB,R3
37 166456 000402      DA500:                      ;* * * * ELSE
38 166460      DA500:                      ;* * * * SET ERROR FLAG
39 166460      DA500:                      ;* * * * SET DONE FLAG
40 166460 052703 100001      BIS   #DCDFGB,R3
41 166460 052703 100001      BIS   #DCDFGB,R3
42 166460 052703 100001      DA600:                      ;* * * * END CASE
43 166464      DA600:                      ;* * END UNTIL
44 166464 032703 000001      BIT   #DCDFGB,R3
45 166464 001750      BEQ   DA100
46 166464 032703 000001      BEQ   DA100
47 166470 001750      DA700:                      ;* * RETURN
48 166472 000207      RETURN
49
50 166472 000207      DA700:                      ;* END GET DATA/COMMAND
51
52
```

```
1           .SBTTL DEPOS - DEPOSITE
2           •ENABLE GBL,LC
3
4
5           ; Module name: DEPOS - DEPOSIT
6           ;
7           ; System: KXT11-CA Native Firmware
8           ;
9           ;
10          ;
11          ;
12          ;
13          ;
14          ; Functional Description:
15          ;
16          ; This module is used by both the Q bus ODT and the serial ODT monitors.
17          ; It deposits the passed data in the register or memory location
18          ; specified.
19
20          ;Input Parameters:
21
22          ;      1) R2 - ADDRESS
23          ;      2) R4 - DATA
24          ;      3) R3 bit REGFGB - Register flag
25
26          ;Output Parameters
27
28          ;      1) R3, BIT (ERRFGB) - ERROR FLAG, BIT = 1 (ERROR), BIT NOT ALTERED IF NO ERROR
29
30          ;DATA STRUCTURES USED
31
32          ;      1) USER REGISTER SAVE AREA
33
34          ;Routines Used
35
36          ;      NONE
37
38          ;Global Symbols Defined
39
40          ;      NONE
41          ;
```

1
2 166474 010246 DEPOS:: ;* PROCEDURE
3 166474 105703 MOV R2,-(SP) ;* * IF TARGET IS A REGISTER
4 166476 100004 TSTB R3
5 166500 BPL DE50
6 166500 100004
7 166502 006302 ASL R2
8 166504 060502 ADD R5,R2
9 166506 062702 177750 ADD #APLR00,R2 ;* * THEN
10 166506 177750 ;* * * CONVERT REGISTER NO. TO ADDRESS
11 166512 DE50: ;* * END IF
12 166512 000241 CLC ;* * DEPOSIT DATA AT SUPPLIED ADDRESS
13 166514 010412 MOV R4,(R2)
14 166516 000240 NOP
15 166520 103002 BCC DF200 ;* * IF NXM
16 166522 052703 100000 BIS #ERPFGB,R3 ;* * THEN
17 166526 DF200: ;* * * SET ERROR FLAG
18 166526 012602 MOV (SP)+,R2 ;* * END IF
19 166530 000207 RETURN ;* * RETURN FROM CALL
20
21
22
23
24
25
26
27
28 ;* END DEPOSIT

```
1 .SBTTL DISPWD - DISPLAY WORD
2 .FENABLE LC,GBL
3
4 ; Module name: DISPWD - DISPLAY WORD
5 ;
6 ; System: KXT11-CA Native Firmware
7 ;
8 ;
9 ;
10 ;
11 ;
12 ;
13 ;
14 ; Functional Description:
15 ;
16 ; This module is a general use utility that converts an 16 bit
17 ; value to an ASCII string and sends it out the console serial
18 ; line.
19 ;
20 ;INPUT PARAMETERS:
21 ;
22 ; 1) R1 - DATA WORD TO BE DISPLAYED
23 ;
24 ;OUTPUT PARAMETERS:
25 ;
26 ; NONE
27 ;
28 ;DATA STRUCTURES USED:
29 ;
30 ; 1) SLU1 CSRs
```

KXTNF - KXT11CA Native Firmware MACRO V05.00 00:51 Page 48
 DISPWD - DISPLAY WORD

75

```

1
2
3 166532           DISPWD::          ;* PROCEDURE DISPLAY WORD
4
5 166532 010046      MOV   R0,-(SP)  ;* * SAVE REGISTERS USED
6 166534 010246      MOV   R2,-(SP)
7
8 166536 012702 000006      MOV   #6,R2  ;* * SET CHARACTER COUNT = 6
9
10
11 166542 005000      CLR   R0
12 166544 006301      ASL   R1
13 166546 005500      ADC   R0
14
15 166550           DI100:          ;* * DO UNTIL CHARACTER COUNT = 0
16
17 166550 062700 000060      ADD   #60,R0  ;* * * CONVERT NEXT NUMBER BUFFER TO ASCII
18
19 166554           DI200:          ;* * * DO UNTIL TRANSMIT BUFFER EMPTY
20
21
22 166554 010046      MOV   R0,-(SP)  ;* * * OR XOFF FLAG = 1
23 166556 004767 005032      CALL  CHKCHR  ;* * * * CHECK FOR INPUT
24 166562 012600      MOV   (SP)+,R0
25
26 166564 105737 177564      TSTB @#XCSR1A  ;* * * END UNTIL
27 166570 100371      BPL   DI200
28 166572 032703 000040      BIT   #XOFFGB,R3
29 166576 001366      BNE   DI200
30
31 166600 110037 177566      MOVB R0,@#XBUF1A  ;* * * DISPLAY NUMBER
32
33 166604 005000      CLR   R0  ;* * * CLEAR NEXT NUBBER BUFFER
34
35
36 166606 006301      ASL   R1  ;* * * SHIFT NEXT THREE HIGH BITS OF DISPLAY
37 166610 005500      ADC   R0  WORD ONTO NEXT NUMBER BUFFER
38 166612 006300      ASL   R0
39 166614 006301      ASL   R1
40 166616 005500      ADC   R0
41 166620 006300      ASL   R0
42 166622 006301      ASL   R1
43 166624 005500      ADC   R0
44
45 166626 005302      DEC   R2  ;* * * DECREMENT CHARACTER COUNT
46
47 166630 001347      BNE   DI100  ;* * END UNTIL
48
49 166632 012602      MOV   (SP)+,R2  ;* * RESTORE REGISTERS
50 166634 012600      MOV   (SP)+,R0
51
52 166636 000207      RETURN        ;* * RETURN FROM CALL
53
                                ;* END DISPLAY WORD

```

```
1 .SBTTL ABAUD - AUTO BAUD
2 .ENABLE LC,CBL
3
4
5 ; Module name: ABAUD - AUTO BAUD
6 ; System: KXT11-CA Native Firmware
7 ;
8 ;
9 ;
10 ;
11 ;
12 ;
13 ; Functional Description:
14 ;
15 ; This module performs the auto baud function on the console serial line.
16 ; It is used by the serial ODT monitor. The decision whether to attempt
17 ; to auto baud is made in this module. If the programmable baud rate
18 ; enable bit of the DLART is set or writing to the baud rate bits
19 ; does not change there value, the auto baud function is not performed.
20 ; If the auto baud function is performed this mode will continue to
21 ; attempt to determine the baud rate until it succeeds.
22 ;
23 ; The auto baud function is performed by setting the baud rate to 2400
24 ; and comparing the input character data to that of a table. This
25 ; character must be a CR and be sent at a baud rate within the range of
26 ; 300 to 9600 (in doubling increments).
27 ;
28 ; The baud rate table has two enteries for 4800 baud to eliminate the
29 ; 4800 baud detection problem that exists for the Falcon.
30 ;
31 ; Input Parameters:
32 ;
33 ; NONE
34 ;
35 ; Output Parameters
36 ;
37 ; NONE
38 ;
39 ; DATA STRUCTURES
40 ;
41 ; SLU1 CSR (XCSR1A)
42 ;
43 ; Routines Used
44 ;
45 ; NONE
46
```

KXTNF - KXT11CA Native Firmware MACRO V05.00 00:51 Page 50
ABAUD - AUTO BAUD

```
1
2      000002          B300  =    2      ;300 BAUD DILART RATE CODE
3      000012          B600  =   12      ;600 " " " "
4      000022          B1200 =   22      ;1200 " " " "
5      000032          B2400 =   32      ;2400 " " " "
6      000042          B4800 =   42      ;4800 " " " "
7      000052          B9600 =   52      ;9600 " " " "
8
9
10 166640 200      INDATA: .BYTE 200      ;RECEIVE DATA AT 300 BAUD/BAUD RATE CODE
11 166641 002      .BYTE B300
12 166642 170      .BYTE 170
13 166643 012      .BYTE B600
14 166644 346      .BYTE 346
15 166645 022      .BYTE B1200
16 166646 015      .BYTE 15
17 166647 032      .BYTE B2400
18 166650 362      .BYTE 362
19 166651 042      .BYTE B4800
20 166652 363      .BYTE 363
21 166653 042      .BYTE B4800
22 166654 377      .BYTE 377
23 166655 052      ENOTAB: .BYTE B9600
24                      .EVEN
```

77

KXTNF - KXT11CA Native Firmware MACRO V05.00 00:51 Page 51
ABAUD - AUTO BAUD

1

```

2 166656                                ABAUD::: ;* PROCEDURE AUTO BAUD
3 166656 032737 000002 177564          BIT    #2,@#XCSR1A ;* * IF DLART XCSR PBRE BIT = 0
4 166664 001041                          BNE    AB600
5
6 166666 052737 000002 177564          BIS    #2,@#XCSR1A ;* * THEN
7                                         BNE    AB600 ;* * * SET PBRE BIT = 1
8
9 166674 032737 000002 177564          BIT    #2,@#XCSR1A ;* * * IF PPRE READS BACK AS 1
10 166702 001432                          BEQ    AB500
11
12 166704 012737 000032 177564          MOV    #B2400,@#XCSR1A ;* * * THEN
13                                         MOV    AB100 ;* * * SET BAUD RATE TO 2400
14
15 166712 005000                         CLR    R0 ;* * * * DO UNTIL BAUD RATE FOUND
16                                         S08    R0,.. ;* * * * DELAY .25 SECONDS
17 166712 005000                         AB100:
18
19 166714 077001                         CLR    R0 ;* * * * CLEAR RECEIVE BUFFER
20                                         S08    R0,.. ;* * * * DO UNTIL DATA RECEIVED
21
22 166716 105737 177562               TSTB   @#RBUF1A ;* * * * FND UNTIL
23
24 166722                               AB200: ;* * * * GET INPUT DATA
25
26 166722 105737 177560               TSTB   @#RCSR1A ;* * * * SET POINTER TO TOP OF TABLE
27 166726 100375                          BPL    AB200
28
29 166730 113700 177562               MOVB   @#RBUF1A,R0 ;* * * * DO WHILE DATA <> (POINTER) OR LAST VALUE
30
31 166734 012701 166640               MOV    #INDATA,R1 ;* * * * INCREMENT POINTER
32
33 166740 120021                         AB300: CMPP   R0,(R1)+ ;* * * * END WHILE
34 166742 001405                         BFEW   AR400
35 166744 022701 166655               CMP    #ENDTAB,R1 ;* * * * FND UNTIL (CONDITION TESTED ABOVE TO SAVE CODE)
36 166750 001760                         BEQ    AR100 ;(NON STRUCTURED BRANCH FOR CODE OPTIMIZATION)
37
38 166752 005201                         INC    R1 ;* * * * SET BAUD RATE PER TABLE
39
40 166754 000771                         BR     AB300 ;* * * * DELAY .5 SECONDS
41
42 166756                               AB400: ;* * * END IF
43
44 166756 111137 177564               MOVR   (R1),@#XCSR1A ;* * END IF
45
46 166762 005000                         CLR    R0
47 166764 077001                         S08    R0,.. ;* * RETURN
48 166766 077001                         S08    R0,.. ;* END AUTO BAUD
49
50 166770                               AB500:
51
52 166770                               AB600:
53
54 166770 000207                         RETURN
55

```

KXTMF - KXT11CA Native Firmware MACRO V05.00 00:51 Page 53
ABAUD - AUTO BAUD

```
1          .SBTTL GOPROC - GO/PROCEED
2          .ENABLE LC,GBL
3
4
5          ;
6          ; Module name: GOPROC - GO/PROCEED
7          ;
8          ; System: KXT11-CA Native Firmware
9          ;
10         ;
11         ;
12         ;
13         ;
14         ;
15         ; Functional Description:
16         ;
17         ; This module performs the go and the proceed functions for both
18         ; the serial and Q bus DDT. It has separate entry points for
19         ; each Go and PROCEED. The difference being that a PC value
20         ; is passed to the module for the GO function. This module
21         ; tests the user stack area and the user PC value. If a NXM is
22         ; encountered at the user SP the module returns from the call.
23         ; If no error is encountered control is passed to the user application.
24
25         ;Input Parameters:
26
27         ;      1) R0 = USER PC VALUE FOR GO COMMAND
28
29         ;Output Parameters
30
31         ;      NONE
32
33         ;Routines Used
34
35         ;      NONE
36
```

```

1
2
3 166772          GO::
4
5
6 166772 042700 000001           BIC   #BIT0,R0
7
8 166776 010065 177766           MOV   R0,APLPCO(R5)
9
10 167002 005065 177770          CLR   APLPSO(R5)
11
12 167006 000005              RESET
13
14 167010          PROC:::
15
16 167010 052737 010000 175002  BIS   #KX$NXP,@#KW.STA
17 167016 016500 177764           MOV   APLSPO(R5),R0
18 167022 012701 052525           MOV   #52525,P1
19 167026 000241
20
21
22 167030 010160 177776           MOV   R1,-2(R0)
23 167034 000240
24 167036 103457           NOP
25 167040 010160 177774           BCS   GP300
26 167044 000240           NOP
27 167046 103453           BCS   GP300
28 167050 020160 177776           CMP   R1,-2(R0)
29 167054 001050           BNE   GP300
30 167056 020160 177774           CMP   R1,-4(R0)
31 167062 001045           BNE   GP300
32
33
34 167064 012700 000000           MOV   #NNCED,R0
35 167070 004767 000104           CALL  SETLED
36
37 167074 042737 010000 175002  BIC   #KX$NXP,@#KW.STA
38
39 167102 016500 177764           MOV   APLSPO(R5),R0
40 167106 016540 177770           MOV   APLPSO(R5),-(R0)
41 167112 016540 177766           MOV   APLPCO(R5),-(R0)
42 167116 010065 177764           MOV   R0,APLSP0(R5)
43
44 167122 012706 177750           MOV   #APLR00,SP
45 167126 060506           ADD   R5,SP
46
47 167130 012600           MOV   (SP)+,R0
48 167132 012601           MOV   (SP)+,R1
49 167134 012602           MOV   (SP)+,R2
50 167136 012603           MOV   (SP)+,R3
51 167140 012604           MOV   (SP)+,R4
52 167142 012605           MOV   (SP)+,R5
53
54 167144 011606           MOV   (SP),SP
55
56 167146 052737 000100 177530  BITS  #KX$DEN,@#KW.CSD
57
;
```

/* PROCEDURE GO/PROCEED
/* * IF GO
/* * THEN
/* * * CLEAR ADDRESS BIT0 OF TARGET
/* * * SET APPLICATION PC = ADDRESS PASSED
/* * * SET APPLICATION PSW = 0
/* * * RESET IOP
/* * END IF
/* * TEST USER SP FOR NXH AND WRITABLE
/* * IF TEST OF USER SP PASSES
/* * THEN
/* * * SET LEDs TO NON NATIVE CODE STATE
/* * * ENABLE NXH'S TO TRAP TO 4
/* * * PUT APPLICATION PC AND PS ON APPLICATION STACK
/* * * SET STACK POINTER TO APPL. R0 SAVE ADDR.
/* * * RESORE USER REGISTERS R0 - R5
/* * RESTORE USER SP
/* * * ENABLE DUAL PORT RAM
/* * * CLEAR SERIAL ODT, QBUS ODT AND NXH FLAGS OF

KXTNF - KXT11CA Native Firmware MACRO V05.00 00:51 Page 54-1
COPROC - GO/PROCEED

```
58 ; STATUS REGISTER
59 167154 042737 010307 175002 BIC #KX$SOF ! KXSQOF ! KX$HXF ! KMSSTF>,@#KW.STA
60 ;* * * SET STATUS REGISTER TO NON NATIVE CODE STATE
61 167162 052737 000007 175002 BIS #KPSNNC,@#KW.STA
62 ;* * * CLEAR COMMAND REGISTER
63 167170 005037 175000 CLR @#KW.CMD
64 ;* * * RETURN TO APPLICATION (NON STRUCTURED EXIT)
65 167174 000002 RTI
66 ;* * * END IF
67 167176 GP300:
68 ;* * * RETURN FROM CALL
69 167176 000207 RETURN
70 ;* END GO/PROCEED
```

KXTNF - KXT11CA Native Firmware MACRO V05.00 00:51 Page 55
SETLED - SET LENS

```
1 .SBTTL SETLED - SET LEDS
2 .ENABLE LC,GBL
3 ;
4 ;
5 ; Module name: SETLED - SET LEDS
6 ;
7 ; System: KXT11-CA Native Firmware
8 ;
9 ;
10 ;
11 ;
12 ;
13 ;
14 ; Functional Description:
15 ;
16 ; This mode sets the LED display to the state pass to it as a parameter.
17 ; It only changes the state of the LEDs if they are not currently
18 ; displaying an error or if the new display is to be Fatal Error.
19 ; There are two entry points to this routine. SETLED is the entry
20 ; used if the return address is on the stack. SETLDJ is used if there
21 ; is not a usable stack and the return address is in R1.
22 ;
23 ;Input Parameters:
24 ;
25 ; 1. New LED State (R0)
26 ; 2. Return address if SETLDJ entry is used (R1)
27 ;
28 ;Output Parameters
29 ;
30 ; NONE
31 ;
32 ;Data structures used
33 ;
34 ; 1. LED State (KXTCSR C, bits 3 - 0)
35 ;
36 ;Routines Used
37 ;
38 ; NONE
39 ;
```

1	000017	LS0	==:	17	;LED STATE 0
2	000016	LS1	==:	16	; " " 1
3	000015	LS2	==:	15	; " " 2
4	000014	LS3	==:	14	; " " 3
5	000013	LS4	==:	13	; " " 4
6	000012	LS5	==:	12	; " " 5
7	000011	LS6	==:	11	; " " 6
8	000010	LS7	==:	10	; " " 7
9	000007	LS8	==:	7	; " " 8
10	000006	LS9	==:	6	; " " 9
11	000005	LS10	==:	5	; " " 10
12	000004	LS11	==:	4	; " " 11
13	000003	LS12	==:	3	; " " 12
14	000002	LS13	==:	2	; " " 13
15	000001	LS14	==:	1	; " " 14
16	000000	LS15	==:	0	; " " 15
17					
18	000017	LEDTSD	==:	LS0	;LED TEST DISPLAY
19	000017	FERPD	==:	LS0	;FATAL ERROR DISPLAY
20	000010	NFERRD	==:	LS7	;NON FATAL ERROR DISPLAY
21	000011	UROMOD	==:	LS6	;USER ROM 0 FAILED DISPLAY
22	000012	UROW1D	==:	LS5	;USER ROM 1 FAILED DISPLAY
23	000013	SILBD	==:	LS4	;SLU1 LOOP BACK TEST FAILED DISPLAY
24	000014	S2ALBD	==:	LS3	;SLU2 CHAN A LOOP BACK TEST FAILED DISPLAY
25	000015	S2BLBD	==:	LS2	;SLU2 CHAN B LOOP BACK TEST FAILED DISPLAY
26	000016	PARLBD	==:	LS1	;PARALLEL LOOP BACK TEST FAILED DISPLAY
27					
28	000007	PUTSDS	==:	LS8	;POWER UP TESTS RUNNING
29	000006	CSTMHD	==:	LS9	;CONTINUOUS SELF TEST MODE DISPLAY
30	000005	QDOTHMD	==:	LS10	;Q BUS ODT MODE DISPLAY
31	000004	SODTHMD	==:	LS11	;SERIAL ODT MODE DISPLAY
32	000003	WAITD	==:	LS12	;WAIT MODE DISPLAY
33	000002	DMAED	==:	LS13	;EXECUTING DMA LOAD DISPLAY
34	000001	PRIPD	==:	LS14	;EXECUTING PRIMARY BOOTSTRAP DISPLAY
35	000000	NNCFD	==:	LS15	;EXECUTING NON NATIVE CODE DISPLAY

1
2
3
4 167200 SETLED:: /* PROCEDURE SET LEDS
5 /* * IF CALLED
6
7 167200 005001 CLR R1 /* * THEN
8 /* * * SET RETURN ADDRESS/FLAG = 0
9 167202 SETLDJ:: /* * END IF
10
11 167202 022700 000017 CMP #FERRD,R0 /* * IF NEW DISPLAY IS FATAL ERROR
12 167206 001405 BEQ SE100 /* * OR LEDS NOT DISPLAYING ERROR
13
14 167210 132737 000010 177524 BITB #KXSLD3,@#KW.CSC
15 167216 001401 BEQ SE100
16 167220 000410 BR SE200
17
18 167222 SE100: /* * THEN
19
20 167222 110037 177524 MOVB R0,@#KW.CSC /* * * SET LED TO STATE PASSED TO ROUTINE (R0)
21
22 167226 122700 000017 CMPB #FERRD,R0 /* * * IF LEDS SET TO FATAL ERROR
23 167232 001003 BNE SE150 /* * * THEN
24
25
26 167234 052737 000040 175002 BYSB #KXSFEF,@#KW.STA /* * * * SET FATAL ERROR FLAG OF STATUS REGISTER
27
28 167242 SE150: /* * END IF
29
30 167242 SE200: /* * END IF
31
32 167242 005701 TST R1 /* * IF RETURN ADDRESS = 0
33 167244 001001 BNE SE300
34
35
36 167246 000207 RETURN /* * * THEN
37
38 167250 SE300: /* * * RETURN FROM CALL
39
40 167250 000111 JMP (R1) /* * ELSE
41
42 /* * * JUMP TO RETURN ADDRESS
/* END SET LEDS

KXTNF - KXT11CA Native Firmware MACRO V05.00 00:51 Page 58
TRAP4 - TRAP 4 EMULATOR

```
1          .SBTTL  TRAP4 - TRAP 4 EMULATOR
2
3          .ENABLE LC,GBL
4
5          ; Module name: TRAP4 - TRAP 4 EMULATOR MODULE
6
7          ; System: KXT11-CA Native Firmware
8
9
10
11
12
13
14          ; Functional Description:
15
16          ; This module emulates the Trap to 4. This emulation
17          ; is performed by placing PSW and the vector address onto the stack
18          ; and performing an RTI. If the stack pointer is not pointing
19          ; to writeable memory (but not NXM) this routine will force a
20          ; stack NXM situation where the Restart Interrupt handler (RESTART)
21          ; will re-establish the stack and retry the trap.
22
```

KXTNF - KXT11CA Native Firmware MACRO V05.00 00:51 Page 59
TRAP4 - TRAP 4 EMULATOR

KXTNF - KXT11CA Native Firmware MACRO V05.00 00:51 Page 60
TRAP10 - TRAP 10 EMULATOR

```
1           .SBTTL TRAP10 - TRAP 10 EMULATOR
2
3           .ENABLE LC,GBL
4
5           ; Module name: TRAP10 - TRAP 10 EMULATOR MODULE
6
7           ; System: KXT11-CA Native Firmware
8
9
10          ;
11          ;
12          ;
13          ;
14          ; Functional Description:
15          ;
16          ; This module emulates the Trap to 10. The trap emulation
17          ; is performed by placing the vector address and PSW on the stack
18          ; and performing an RTI. If the stack pointer is not pointing
19          ; to writable memory (but not NXM) this routines will force a
20          ; stack NXM situation where the Restart Interrupt handler (RESTART)
21          ; will re-establish the stack and retry the trap.
22
```

85

KXTNF - KXT11CA Native Firmware MACRO V05.00 00:51 Page 61
TRAP10 - TRAP 10 EMULATOR

```
1           ;* PROCEDURE TRAP4
2 167314          TRAP10::             ;* * SET UP STACK FOR TRAP TO 4
3
4 167314  013746  000012      MOV    @#12,-(SP)
5 167320  013746  000010      MOV    @#10,-(SP)
6
7 167324  023766  000012  000002  CMP    @#12,(SP)
8 167332  001004          BNE    TR1010
9 167334  023716  000010          CMP    @#10,(SP)
10 167340  001001          BNE    TR1010
11
12          ;* * THEN
13 167342  000002          RTI    ;* * * EMULATE TRAP (NON STRUCTURED EXIT)
14
15 167344          TR1010::            ;* * ELSE
16
17 167344  052737  004000  175002  BIS    #KXSSXT,@NWKW.STA
18 167352  005737  177776          TST    @#177776 ;(FOPCE A NXM RESTART INT.)
19
20          ;* * END IF
          ;* END TRAP10
```

KXTNF - KXT11CA Native Firmware MACRO V05.00 00:51 Page 62
TRAP24 - TRAP 24 EMULATOR

```
1           .SBTTL TRAP24 - TRAP 24 EMULATOR
2
3           .ENABLE LC,GBL
4
5           ; Module name: TRAP24 - TRAP 24 EMULATOR MODULE
6
7           ; System: KXT11-CA Native Firmware
8
9           ;
10          ;
11          ;
12          ;
13          ;
14          ; Functional Description:
15
16          ; This module emulates the Trap to 24. This emulation
17          ; is performed by placing data at locations 26 into the PSW
18          ; and 24 int the PC.
19
```

88
51

KXTNF - KXT11CA Native Firmware MACRO V05.00 00:51 Page 63
TRAP24 - TRAP 24 EMULATOR

```
1           TPAP24::                      /* PROCEDURE TRAP24
2 167356
3
4 167356 106437 000026      MTPS    @#26      /* * MOVE CONTENTS OF 26 TO PSW
5
6 167362 013707 000024      MOV     @#24,PC   /* * MOVE CONTENTS OF 24 TO PC
7
8
9
```

```
1 .SBTTL TRAPX - TRAP COMMAND EMULATOR
2
3 .ENABLE LC,GBL
4 ;
5 ; Module name: TRAPX - TRAP COMMAND EMULATOR MODULE
6 ;
7 ; System: KXT11-CA Native Firmware
8 ;
9 ;
10 ;
11 ;
12 ;
13 ;
14 ; Functional Description:
15 ;
16 ; This module emulates the Trap specified by the DPR. This emulation
17 ; is performed by placing data in DPR register 2 into the PSW
18 ; and DPR register 3 into the PC.
19 ;
20 ;* PROCEDURE TRAPX
21 167366 TRAPX::: ;* * CLEAR COMMAND REGISTER
22
23 167366 005037 175000 CLR @#KW.CMD ;* * SET PSW TO DPR REGISTER 3
24
25 167372 106437 175006 MTPS @#KW.SC3 ;* * JUMP TO ADDRESS SPEC. IN DPR REGISTER 2
26
27 167376 013707 175004 MOV @#KW.SC2,PC ;* END TRAPX
28
29
```

1 .SBTTL WAITST - WAIT FOR COMMAND STATE
2 .ENABLE LC,GBL
3
4 ;
5 ; Module name: WAITST
6 ;
7 ; System: KXT11-CA Native Firmware
8 ;
9 ;
10 ;
11 ;
12 ;
13 ;
14 ; Functional Description:
15 ;
16 ; This module is used to place the IOP in the wait state. In this state
17 ; the IOP waits for a command to be received from the Q bus.
18 ; The LED if not displaying an error will be set to the wait state
19 ; display. The state field of status register of the DPR will be
20 ; set to the wait state if the Root/Self Test Switch is not set to 10.
21 ; If the switch is set to 10 then the state field is set to the Q Bus
22 ; controlled test state. The routine continues to loop setting the LEDs
23 ; and the state field of the status register so that if the serial ODT
24 ; mode is entered via Break, while in this routine and a Proceed
25 ; command is executed the LED display and state field will be correct.
26 ;
27 ; Input Parameters:
28 ;
29 ; ; NONE
30 ;
31 ; Output Parameters
32 ;
33 ; ; NONE
34 ;
35 ; Routines Used
36 ;
37 ; ; NONE

KXTNF - KXT11CA Native Firmware MACRO V05.00 00:51 Page 66
 WAITST - WAIT FOR COMMAND STATE

```

1
2 167402
3
4 167402
5
6 167402 012700 000003
7 167406 004767 177566
8
9 167412 013700 175002
10 167416 042700 000007
11
12 167422 113701 177522
13 167426 142701 177417
14 167432 122701 000240
15 167436 001003
16
17
18
19 167440 052700 000002
20 167444 000402
21 167446
22
23
24 167446 052700 000004
25
26 167452
27
28 167452 010037 175002
29
30 167456 000751
31

        WAITST:::                                ;* PROCEDURE WAITST
                                                ;* * DO UNTIL INTERRUPTED
                                                ;* * * SET LED DISPLAY TO WAIT STATE
                                                ;* * * GET STATUS REGISTER
                                                ;* * * IF THE BOOT/SELF TEST SWITCH = 10
                                                ;* * * THEN
                                                ;* * * * SET STATE FIELD OF STATUS REG. TO
                                                ;* * * * QBUS CONTROLLED TEST STATE
                                                ;* * * ELSE
                                                ;* * * * SET STATE FIELD OF STATUS REG. TO WAIT STATE
                                                ;* * * END IF
                                                ;* * * WRITE STATUS REGISTER DATA
                                                ;* * END UNTIL
                                                ;* END WAITST

        WA100:                                MOV    #WAITD,R0
                                                CALL   SETLED
                                                MOV    @#KW.STA,R0
                                                BIC    #KMSSTF,R0
                                                MOVB  @#KL.CSB,P1
                                                BICB  #<KMSWS>,R1
                                                CMPB  #<10.*16.>,R1
                                                BNE   WA200
                                                BTS   #KPSQTS,R0
                                                BR    WA300
        WA200:                                BIS   @KPSWST,R0
                                                WA300:                               
                                                MOV    R0,@#KW.STA
                                                BR    WA100

```

```
1 .SBTTL QCOMIN - Q BUS COMMAND INTERPP.  
2 .ENABLE GBL,LC  
3  
4 ;  
5 ; Module name: QCOMIN - Q BUS COMMAND INTERPRITOR  
6 ;  
7 ; System: KXT11-CA Native Firmware  
8 ;  
9 ;  
10 ;  
11 ;  
12 ;  
13 ;  
14 ; Functional Description:  
15 ;  
16 ; This module decodes the Q bus commands received through the command  
17 ; register of the dual port RAM. Once a valid command is recognised  
18 ; the appropriate routine is called. If the command received is a test  
19 ; command the test command decode routine (DECTST) is jumped to.  
20 ;  
21 ;Input Parameters:  
22 ;  
23 ;      NONE  
24 ;  
25 ;Output Parameters:  
26 ;  
27 ;      NONE  
28 ;  
29 ;Data Structures Used  
30 ;  
31 ;      1. IOP COMMAND REGISTER  
32 ;  
33 ;      2. IOP STATUS REGISTER  
34 ;  
35 ;Routines Used  
36 ;  
37 ;      1. REINIT (JMP)  
38 ;  
39 ;      2. DECTST (JMP)  
40 ;  
41 ;      3. WAITST (JMP)
```

```

1                                         /* PROCEDURE Q BUS COMMAND INTERPRETER
2 167460                                     QCOMIN:::
3
4
5 167460 042737 140000 175002             BIC   #<KXSCME ! KXSDME>, @#KW.STA
6                                         ;* * CLEAR COMMAND ERROR AND DMA LOAD ERROR
7 167466 032737 000100 175002             BIT   #KX$SOF, @#KW.STA
8 167474 001406                           BEQ   QC50
9                                         ;* * THEN
10                                         ;* * * SET COMMAND ERROR BIT OF STATUS REGISTER
11 167476 052737 100000 175002             BIS   #KXSCME, @#KW.STA
12                                         ;* * * CLEAR COMMAND REGISTER
13 167504 005037 175000                  CLR   @#KW.CMD
14                                         ;* * * RETURN FROM INTERRUPT
15 167510 000002                           RTI   ;(NON STRUCTURED EXIT)
16                                         ;* * END IF
17 167512                                     QC50:
18                                         ;* * IF QODT FLAG OF STATUS REGISTER = 1
19 167512 032737 000200 175002             BIT   #KXSQOF, @#KW.STA
20 167520 001401                           BEQ   QC70
21                                         ;* * THEN
22                                         ;* * * UNBLOCK QODT WAIT FOR COMMAND
23 167522 000002                           RTI   ;(NON STRUCTURED EXIT)
24                                         ;* * END IF
25 167524                                     QC70:
26                                         ;* * IF TEST/COMMAND BIT = 0
27 167524 032737 100000 175000             BIT   #BIT15, @#KW.CMD
28 167532 001147                           BNE   QC500
29                                         ;* * THEN
30                                         ;* * * CASE COMMAND
31                                         ;* * * WHEN Q BUS ODT
32 167534 022737 000010 175000             CMP   #KXSODC, @#KW.CMD
33 167542 001006                           BNE   QC100
34                                         ;* * * AND STATUS REGISTER SERIAL ODT FLAG = 0
35 167544 032737 000100 175002             BIT   #KX$SOF, @#KW.STA
36 167552 001002                           BNE   QC100
37                                         ;* * * * ENTER ODT Q BUS ODT MODE
38                                         ;* * * * (NON STRUCTURED JUMP TO Q BUS ODT MONITOR)
39 167554 000167 001652                  JMP   QODTM
40                                         ;* * * WHEN RE-INITIALIZE
41 167560 022737 000004 175000             CMP   #KXSINC, @#KW.CMD
42 167566 001024                           BNE   QC200
43                                         ;* * * AND DPR REGISTER 3 = 0 THROUGH 6 AND ID
44                                         ;* * * SWITCH < 7
45 167570 022737 000006 175006             CMP   #6, @#KW.SC3
46 167576 103407                           BLO   QC120
47 167600 013700 177522                  MOV   @#KW.CSB, R0
48 167604 042700 177417                  BIC   #^C<KMSSWS>, R0
49 167610 022700 000160                  CMP   #<7*16.>, R0
50 167614 101004                           BHI   QC150
51                                         ;* * * OR DPR REGISTER 3 = 8
52 167616                                     QC120:
53 167616 022737 000010 175006             CMP   #8., @#KW.SC3
54 167624 001005                           BNE   QC200
55 167626                                     QC150:
56                                         ;* * * * DISABLE DUAL PORT PAM
57 167626 042737 000100 177520             BIC   #KXSDFN, @#KW.CSA

```

```

58 ;* * * * RE-INITIALIZE
59 ; (NON STRUCTURED JUMP TO POWERUP MODULE RE-INITIALIZE
60 167634 000167 003156 JMP REINIT
61 ;* * * WHEN DMA LOAD
62 167640 022737 000002 175000 QC200: CMP #KXSDMC,@#KW.CMD
63 167646 001004 BNE QC300
64 ;* * * REMOVE RETURN ADDR. AND PS FROM STACK
65 167650 062706 000004 ADD #4,SP
66 ;* * * PERFORM DMA TRANSFER
67 ; (NON STRUCTURED JUMP TO DMA LOADER MODULE)
68 167654 000167 001152 JMP DMALD
69 ;* * * WHEN TRAP COMMAND
70 167660 022737 000001 175000 QC300: CMP #KXSTRC,@#KW.CMD
71 167666 001023 BNE QC400
72 ;* * * IF PC VALUE IN DPR REGISTER 2 IS EVEN
73 167670 032737 000001 175004 BIT #8ITO,@#KW.SC2
74 167676 001016 BNE QC380
75 ;* * * THEN
76 ;* * * SET LEDS TO NON NATIVE CODE STATE
77 167700 012700 000000 MOV #NNCED,R0
78 167704 012701 167714 MOV #QC350,R1
79 167710 000167 177266 JMP SFTLQJ
80 167714 QC350: ;* * * * SET STATUS REGISTER NON NATIVE CODE STATE
81 82 167714 042737 000007 175002 BIC #KMSSTF,@#KW.STA
83 167722 052737 000007 175002 BITS #KPSNNC,@#KW.STA
84 ;* * * * EMULATE TRAP AT ADDRESS
85 ; PER DUAL PORT REGISTER 2
86 ; SET PSW PER DUAL PORT REGISTER 3
87 ; (NON STRUCTURED EXIT)
88 167730 000167 177432 JMP TRAPX
89 ;* * * END IF
90 167734 000442 QC380: BR QC450
91 ;* * * WHEN SHOW STATUS COMMAND
92 167736 QC400: ;* * * * PLACE BOOT SWITCH AND MAP SWITCH
93 167736 022767 000020 005034 CNP #KXSSHG,KW.CMD
94 167744 001027 BNE QC447
95 ;* * * VALUES TO DPR REGISTER 3
96 ;* * * * CLEAR ALL NON RELATED BITS
97 167746 113737 177522 175006 MOVB @#KL.CSB,@#KL.SC3
98 ;* * * * IF USER SOCKETS MAPPED LOW
99 167754 042737 177401 175006 BIC #^CCKM$MAP I KMSSMS>,@#KW.SC3
100 101 167762 032737 000010 177522 BIT #KXSMP2,@#KL.CSB
102 167770 001402 BEQ QC420
103 ;* * * THEN
104 ;* * * * TEST FOR RAM AT 0
105 167772 005000 CLR R0
106 167774 000402 BR QC440
107 ;* * * ELSE
108 167776 QC420: ;* * * * TEST FOR RAM AT 100000
109 110 167776 012700 100000 MOV #100000,R0
111 ;* * * END IF
112 170002 QC440: ;* * * * IF RAM PRESENT
113
114 170002 011001 MOV (R0),R1

```

```

    115 170004 005110      COM   (R0)
    116 170006 020110      CMP   R1,(R0)
    117 170010 001404      BEQ   QC445
    118                                     ;* * * * THEN
    119                                     ;* * * * RESTORE LOCATION
    120 170012 010110      MOV   R1,(R0)
    121                                     ;* * * * SET RAM PRESENT BIT OF DPR REG. 3
    122 170014 052737 000001 175006      BIS   #BIT0,@#KW.SC3
    123                                     ;* * * * END IF
    124 170022              QC445:
    125 170022 000412      BR    QC470
    126                                     ;* * * WHEN NOP COMMAND
    127 170024              QC447:
    128 170024 022737 000040 175000      CMP   @#KW.CMD
    129 170032 001003      BNE   QC450
    130                                     ;* * * CLR FAR COMMAND REGISTER
    131 170034 005037 175000      CLR   @#KW.CMD
    132                                     ;* * * RETURN FROM INTERRUPT
    133 170040 000002      RTI   ;(NON STRUCTURED EXIT)
    134                                     ;* * * ELSE (BAD COMMAND)
    135 170042              QC450:
    136                                     ;* * * SET COMMAND ERROR BIT OF STATUS REGISTER
    137 170042 052737 100000 175002      BTS   #KX$CME,@#KW.STA
    138                                     ;* * * END CASE
    139 170050              QC470:
    140 170050 000402      BR    QC600
    141                                     ;* * ELSE (TEST/COMMAND BIT = 1)
    142 170052              QC500:
    143                                     ;* * * DECODE TEST COMMAND (NON-STRUCTURED JUMP TO DECTST)
    144 170052 000167 000014      JMP   DECTST
    145                                     ;* * END IF
    146 170056              QC600:
    147                                     ;* * REMOVE RETURN ADDRESS AND PSW FROM STACK
    148 170056 062706 000004      ADD   #4,SP
    149                                     ;* * CLEAR COMMAND REGISTER
    150 170062 005037 175000      CLR   @#KW.CMD
    151                                     ;* * ENTER WAIT STATE (NON STRUCTURED EXIT)
    152 170066 000167 177310      JMP   WAITST
    153                                     ;* END Q BUS COMMAND INTERPRETER

```

1 .S3TTL DECTST - DECODE AND DISPATCH QBUS SELF TEST REQUESTS
2 .ENABL GRL
3
4
5 ; MODULE NAME: DECTST - DECODE AND DISPATCH QBUS SELF TEST REQUESTS
6 ;
7 ;
8 ;
9 ;
10 ;
11 ;
12 ;
13 ; FUNCTIONAL DESCRIPTION:
14 ;
15 ; THIS MODULE DECODES THE QBUS SELF TEST REQUESTS AND DISPATCHES TO THE
16 ; APPROPRIATE SRLF TEST MODULE. ON RETURN FROM THE SELF TEST MODULE,
17 ; DECTST UPDATES THE LED DISPLAY, UPDATES THE ACCUMULATED ERROR WORD,
18 ; AND PERFORMS THE ACKNOWLEDGEMENT PORTION OF THE PROTOCOL TO NOTIFY THE
19 ; ARBITER OF COMPLETION.
20 ;
21 ; INPUTS:
22 ; TPR 0 CONTAINS THE SELF TEST REQUEST CODE.
23 ; TPR 3 CONTAINS AN OPTIONAL CALLING PARAMETER, IF ANY.
24 ;
25 ; CALLING SEQUENCE:
26 ; A SPECIAL CASE JMP TO DECTST FROM QCOMIN.
27 ;
28 ; OUTPUTS:
29 ; TPR 0 IS CLEAR.
30 ; TPR 1 IS THE SAME AS WHEN CALLED UNLESS DECTST SETS THE COMMAND ERROR
31 ; BIT, BIT 15. IF THE BOOT SWITCH SELECTION IS 10 (QBUS
32 ; CONTROLLED TEST), THE STATE FIELD IS SET TO "WAITING FOR
33 ; QBUS TEST COMMAND". ELSE, THE STATE FIELD IS SET TO "WAITING
34 ; FOR QBUS COMMAND".
35 ; TPR 2 HAS ITS APPROPRIATE TEST BIT CLEARED IF NO ERROR OCCURRED
36 ; AND SET IF AN ERROR OCCURRED. TPR 2 IS THE ACCUMULATED SELF
37 ; TEST RESULTS.
38 ; IF TPR 2'S APPROPRIATE ERROR BIT IS SET, TPR 3 CONTAINS THE TEST CODE
39 ; IN BITS <15:12> AND THE DISCRETE ERROR CODE IN BITS <11:00>.
40 ;
41 ; IF AN ERROR OCCURRED, THE LEDS ARE UPDATED WITH A CALL TO TNERR.
42 ;
43 ; ROUTINES USED:
44 ; SPR5SU, TNERR, & THE VARIOUS SELF TEST ROUTINES
45 ;
46 ; OTHER NOTES AND COMMENTS:
47 ;
48 ; 1. THE ARBITER IS RESPONSIBLE FOR CLEARING THE ACCUMULATED VALUE IN
49 ; TPR 2.
50 ;
51 ; 3. ALL REGISTERS ARE USEABLE WITHOUT RESTORATION.

```

1 170072          DECTST::      MOV     #3$ ,R0      ;SET UP A RETURN ADDRESS FROM SPR5SU.
2 170072 012700 170102      JMP     @#SPR5SU     ;GO SET UP R5 AND A DEFAULT STACK POINTER.
3 170076 000137 173706
4 170102
5 170102 012700 170354      3$:      MOV     #STSTAB,R0      ;POINT TO THE SELF TEST DISPATCH TABLE.
6 170106 005710      TST     (R0)      ;ARE WE AT THE END OF THE TABLE?
7 170110 001005      BNE     10$      ;NO.
8 170112 013700 175002      MOV     @#KW.STA,R0      ;YES, GET TPR 1.
9 170116 052700 100000      BIS     #KXSCME,R0      ;SET THE COMMAND ERROR BIT IN THE SAVED TPR 1.
10 170122 000465      BR      45$      ;GO HANDLE THE REST OF UPDATING TPR 1 & TPR 0.
11 170124 021037 175000     10$:     CMP     (RC),@#KW.CMD   ;NO, DOES THE TABLE ENTRY MATCH TPR 0?
12 170130 001403      BEQ     15$      ;YES.
13 170132 062700 000010      ADD     #SETSENL,R0      ;NO, ADD THE LENGTH OF A TABLE ENTRY TO THE
14                           ;POINTER.
15 170136 000763
16 170140
17 170140 013746 175002      15$:     BR      5$       ;GO TRY THE NEXT ENTRY IN THE TABLE.
18 170144 013746 175004      MOV     @#KW.STA,-(SP)  ;WE HAVE FOUND A VALID COMMAND, SO SAVE
19 170150 012046      MOV     @#KW.SC2,-(SP)  ;TPR 1 (THE STATUS REGISTER),
20 170152 042716 100000      MOV     (R0)+,-(SP)   ;TPP 2,
21 170156 012046      BTC     #BIT15,(SP)   ;THE COMMAND (BIT 15 CLEARED),
22                           ;THE ERROR CONDITION ROUTINE'S ADDRESS ON
23 170160 013746 175006      MOV     @#KW.SC3,-(SP)  ;THE STACK,
24 170164 013746 177524      MOV     @#KW.CSC,-(SP) ;THE LEDS.
25 170170 005001      CLR     R1       ;CLEAR R1 TO DEFAULT TO NO PARAMETER IN TPR 3.
26 170172 105720      TSTB    (R0)+      ;NOW, IS THERE A PARAMETER IN TPR 3?
27 170174 001402      BFQ     20$      ;NO, R1 HAS ALREADY BEEN CLEARED.
28 170176 013701 175006      MOV     @#KW.SC3,R1      ;YES, PUT THE PARAMETER IN R1.
29 170202 105720
30 170204 001404      20$:     TSTB    (R0)+      ;IS THE TEST ROUTINE IN OVERLAY 0?
31 170206 052737 000040 177520      BIS     #KXSDPE,@#KW.CSA ;NO, IT'S IN OVERLAY 1, SO ENABLE THE
32                           ;OVERLAY 1 MAP.
33 170214 000403
34 170216 042737 000040 177520      25$:     BR      30$      ;IT'S IN OVERLAY 0, SO DISABLE THE
35                           ;OVERLAY 1 MAP.
36 170224 052737 010000 175002      BIC     #KXSNXF,@#KW.STA ;ENABLE C-BIT SETTING ON REFERENCES TO
37                           ;NON-EXISTENT MEMORY (RATHER THAN A TRAP THRU
38                           ;#4). KXSNXF WILL BE RESTORED WHEN TPR 1 IS
39                           ;RESTORED.
40 170232 004730      JSR     PC,@(R0)+      ;CALL THE TEST ROUTINE VIA THE "VECTOR".
41 170234 012637 177524      MOV     (SP)+,@#KW.CSC ;RESTORE THE LEDS.
42 170240 005700      TST     R0       ;WAS AN ERROR DETECTED?
43 170242 001406      BEQ     35$      ;NO.
44 170244 010037 175006      MOV     R0,@#KW.SC3      ;YES, SET TPR 3 FOR THIS ERROR.
45 170250 012601      MOV     (SP)+,R1      ;PUT THE VALUE OF TPR 3, WHEN CALLED, IN R1.
46                           ;(IN CASE THE LED HANDLING ROUTINE NEEDS IT).
47 170252 004736      JSR     PC,@(SP)+      ;YES, GO TO THE LED UPDATING ROUTINE.
48 170254 052616      BIS     (SF)+,(SP)   ;SET THE TEST'S ERROR BIT IN THE SAVED TPR 2.
49 170256 000404      BR      40$      ;GO FINISH UP TPR 2, 1, AND 0 UPDATING.
50 170260 012637 175006      35$:     MOV     (SP)+,@#KW.SC3 ;RESTORE TPR 3.
51 170264 005726      TST     (SP)+      ;FLUSH THE ERROR HANDLER ADDRESS OFF THE
52                           ;STACK.
53 170266 042616      BIC     (SP)+,(SP)   ;CLEAR THE TEST'S ERROR BIT IN THE SAVED
54                           ;TPR 2.
55 170270 012637 175004      40$:     MOV     (SP)+,@#KW.SC2 ;STORE THE RESULT IN TPR 2.
56 170274 012600      MOV     (SP)+,R0      ;GET THE SAVED TPR 1.
57 170276 042700 000007      45$:     BTC     #KMSSTF,R0      ;CLEAR THE SAVED STATE FIELD.

```

KXTNF - KXT11CA Native Firmware MACRO V05.00 00:51 Page 70-1
 DECTST - DECODE AND DISPATCH QBUS SELF TEST REQUESTS

```

58 170302 013701 177522      MOV    @#KW.CSB,R1      ;GET THE BOOT SWITCH SETTING FROM KXTCSPB BITS
59
60 170306 042701 177417      BIC    #^C<KM$SWS>,R1  ;ISOLATE THE SWITCH SETTING BITS.
61 170312 022701 000240      CMP    #10.*20,R1      ;IS THE SWITCH SETTING FOR "QBUS CONTROLLED
62
63 170316 001003
64 170320 052700 000002      BNE    47S
65
66 170324 000402
67 170326 052700 000004      47S:   BIS    #KPS#ST,RO      ;SET THE STATE TO "WAITING FOR QBUS COMMAND".
68 170332 010037 175002      50$:   MOV    R0,@#KW.STA     ;RESTORE THE STATUS REGISTER.
69 170336 005037 175000      CLR    @#KW.CMD      ;CLEAR THE COMMAND REGISTER, TPR 0.
70 170342 052737 000100 177530 BIS    #KXSDEN,@#KW.CSD ;RE-FNABLE THE TWO PORT RAM (SOMETIMES IT'S
71
72
73 170350 000137 167402      JMP    @#WAITST      ;NEEDED, SOMETIMES IT'S NOT, BUT DO IT
74
75
76 ;STSTAB, THE SELF TEST DISPATCH TABLE, DEFINES THE LEGAL SELF TEST CODES, THE
77 ;EXISTENCE OF A CALLING PARAMETER IN TPR 3, THE "ROM PAGE" CONTAINING THE TEST
78 ;CODE, AND THE ADDRESS OF THE CODE.
79 ;
80 ;     .WORD    TPR 0 VALUE OR 0      ;TPR 0 VALUE IF LEGAL DISPATCH CODE &
81 ;     ;          0 IF END OF LIST (THUS IMPLYING
82 ;     ;          COMMAND ERROR)
83 ;
84 ;     .WORD    TNERR               ;ADDRESS OF ROUTINE THAT HANDLES LED
85 ;     ;          UPDATING IF AN ERROR IS INDICATED.
86 ;
87 ;     .BYTE    0 OR NON-0         ;NON-0 IMPLIES THAT A CALLING
88 ;     ;          PARAMETER IS IN TPR 3. 0
89 ;     ;          IMPLIES NO PARAMETER IN TPR 3.
90 ;
91 ;     .BYTE    0 OR NON-0         ;NON-0 IMPLIES THAT THE TEST ROUTINE
92 ;     ;          IS IN THE HIGH OVERLAY, OVERLAY 1, OF
93 ;     ;          ROM, THUS IMPLYING THAT KXTCSRA
94 ;     ;          BIT 5 MUST BE 1 TO MAP THE CODE
95 ;     ;          TO THE 160000 BASE ADDRESS. 0
96 ;     ;          IMPLIES THAT THE TEST ROUTINE IS
97 ;     ;          IN THE LOW OVERLAY, OVERLAY 0,
98 ;     ;          OF ROM, THUS KXTCSRA BIT 05 MUST
99 ;     ;          BE 0 TO MAP THE CODE TO THE
100 ;          ;          160000 BASE ADDRESS.
101 ;
102 ;     .WORD    ADDRESS            ;THE ADDRESS OF THE ROUTINE'S "VECTOR"
103 ;     ;          IN THE FIRST 32 (DECIMAL) BYTES OF
104 ;     ;          THE ROM OVERLAY.
105 ;
106 170354
107 ;STSTAB:
;CSR TEST
108 170354 100001      .WORD   KX$T01      ;TPR 0 VALUE
109 170356 170506      .WORD   T1ERR      ;LED UPDATING ROUTINE ON ERROR
110 170360 000          .BYTE   0          ;NO INPUT PARAMETER
111 170361 000          .BYTE   0          ;IN ROM OVERLAY 0
112 170362 160004      .WORD   T1          ;VECTOR ADDRESS
113
114 000010             STSENL =  .-STSTAB    ;DEFINE THE TABLE ENTRY LENGTH

```

97

```

115
116
117 170364 100002           ;RAM TEST
118 170366 170546           .WORD   KX$T02
119 170370 000               .WORD   T2ERR
120 170371 000               .BYTE   0
121 170372 160010           .BYTE   0
122
123 170374 100004           .WORD   KX$T03
124 170376 170576           .WORD   T3ERR
125 170400 001               .BYTE   1
126 170401 000               .BYTE   0
127 170402 160014           .WORD   T3
128
129 170404 100010           ;ROM TEST
130 170406 170642           .WORD   KX$T04
131 170410 000               .WORD   T4ERR
132 170411 000               .BYTE   0
133 170412 160020           .BYTE   0
134
135 170414 100020           .WORD   KX$T05
136 170416 170650           .WORD   T5ERR
137 170420 000               .BYTE   0
138 170421 000               .BYTE   0
139 170422 160024           .WORD   T5
140
141 170424 100040           ;CPU TEST
142 170426 170656           .WORD   KX$T06
143 170430 000               .WORD   T6ERR
144 170431 000               .BYTE   0
145 170432 160030           .BYTE   0
146
147 170434 100100           .WORD   KX$T07
148 170436 170664           .WORD   T7ERR
149 170440 001               .BYTE   1
150
151 170441 000               .BYTE   0
152 170442 160034           .WORD   T7
153
154 170444 100200           ;BEVENT TEST
155 170446 170720           .WORD   KX$T08
156 170450 000               .WORD   T8ERR
157 170451 001               .BYTE   0
158 170452 160000           .BYTE   1
159
160 170454 100400           .WORD   KX$T09
161 170456 170726           .WORD   T9ERR
162 170460 001               .BYTE   1
163
164 170461 001               ;SLU 1 TEST
165 170462 160004           .WORD   KX$T10
166
167 170464 101000           .WORD   T10ERR
168 170466 170734           .BYTE   0
169 170470 001               .BYTE   1
170
171 170471 001               ;SLU 2 TEST
172
173 170474 101100           .WORD   KX$T11
174 170476 170736           .WORD   T11ERR
175 170478 001               .BYTE   1
176
177 170481 000               ;PIO TEST
178 170483 160008           .WORD   KX$T12
179 170485 001               .WORD   T12FRR
180 170487 000               .BYTE   1
181
182 170489 001               ;DMA TEST
183 170491 160012           .WORD   KX$T13
184 170493 001               .WORD   T13ERR
185 170495 000               .BYTE   1
186
187 170497 001               ;QIR TEST
188 170499 160016           .WORD   KX$T14
189 170501 001               .WORD   T14FRR
190 170503 000               .BYTE   1
191
192 170505 001               ;INTERRUPT TEST
193 170507 160020           .WORD   KX$T15
194 170509 001               .WORD   T15FRR
195 170511 000               .BYTE   1
196
197 170513 001               ;ROM TEST
198 170515 160024           .WORD   KX$T16
199 170517 001               .WORD   T16FRR
200 170519 000               .BYTE   1
201
202 170521 001               ;ROM TEST
203 170523 160028           .WORD   KX$T17
204 170525 001               .WORD   T17FRR
205 170527 000               .BYTE   1
206
207 170529 001               ;ROM TEST
208 170531 160032           .WORD   KX$T18
209 170533 001               .WORD   T18FRR
210 170535 000               .BYTE   1
211
212 170537 001               ;ROM TEST
213 170539 160036           .WORD   KX$T19
214 170541 001               .WORD   T19FRR
215 170543 000               .BYTE   1
216
217 170545 001               ;ROM TEST
218 170547 160040           .WORD   KX$T20
219 170549 001               .WORD   T20FRR
220 170551 000               .BYTE   1
221
222 170553 001               ;ROM TEST
223 170555 160044           .WORD   KX$T21
224 170557 001               .WORD   T21FRR
225 170559 000               .BYTE   1
226
227 170561 001               ;ROM TEST
228 170563 160048           .WORD   KX$T22
229 170565 001               .WORD   T22FRR
230 170567 000               .BYTE   1
231
232 170569 001               ;ROM TEST
233 170571 160052           .WORD   KX$T23
234 170573 001               .WORD   T23FRR
235 170575 000               .BYTE   1
236
237 170577 001               ;ROM TEST
238 170579 160056           .WORD   KX$T24
239 170581 001               .WORD   T24FRR
240 170583 000               .BYTE   1
241
242 170585 001               ;ROM TEST
243 170587 160060           .WORD   KX$T25
244 170589 001               .WORD   T25FRR
245 170591 000               .BYTE   1
246
247 170593 001               ;ROM TEST
248 170595 160064           .WORD   KX$T26
249 170597 001               .WORD   T26FRR
250 170599 000               .BYTE   1
251
252 170601 001               ;ROM TEST
253 170603 160068           .WORD   KX$T27
254 170605 001               .WORD   T27FRR
255 170607 000               .BYTE   1
256
257 170609 001               ;ROM TEST
258 170611 160072           .WORD   KX$T28
259 170613 001               .WORD   T28FRR
260 170615 000               .BYTE   1
261
262 170617 001               ;ROM TEST
263 170619 160076           .WORD   KX$T29
264 170621 001               .WORD   T29FRR
265 170623 000               .BYTE   1
266
267 170625 001               ;ROM TEST
268 170627 160080           .WORD   KX$T30
269 170629 001               .WORD   T30FRR
270 170631 000               .BYTE   1
271
272 170633 001               ;ROM TEST
273 170635 160084           .WORD   KX$T31
274 170637 001               .WORD   T31FRR
275 170639 000               .BYTE   1
276
277 170641 001               ;ROM TEST
278 170643 160088           .WORD   KX$T32
279 170645 001               .WORD   T32FRR
280 170647 000               .BYTE   1
281
282 170649 001               ;ROM TEST
283 170651 160092           .WORD   KX$T33
284 170653 001               .WORD   T33FRR
285 170655 000               .BYTE   1
286
287 170657 001               ;ROM TEST
288 170659 160096           .WORD   KX$T34
289 170661 001               .WORD   T34FRR
290 170663 000               .BYTE   1
291
292 170665 001               ;ROM TEST
293 170667 160100           .WORD   KX$T35
294 170669 001               .WORD   T35FRR
295 170671 000               .BYTE   1
296
297 170673 001               ;ROM TEST
298 170675 160104           .WORD   KX$T36
299 170677 001               .WORD   T36FRR
300 170679 000               .BYTE   1
301
302 170681 001               ;ROM TEST
303 170683 160108           .WORD   KX$T37
304 170685 001               .WORD   T37FRR
305 170687 000               .BYTE   1
306
307 170689 001               ;ROM TEST
308 170691 160112           .WORD   KX$T38
309 170693 001               .WORD   T38FRR
310 170695 000               .BYTE   1
311
312 170697 001               ;ROM TEST
313 170699 160116           .WORD   KX$T39
314 170701 001               .WORD   T39FRR
315 170703 000               .BYTE   1
316
317 170705 001               ;ROM TEST
318 170707 160120           .WORD   KX$T40
319 170709 001               .WORD   T40FRR
320 170711 000               .BYTE   1
321
322 170713 001               ;ROM TEST
323 170715 160124           .WORD   KX$T41
324 170717 001               .WORD   T41FRR
325 170719 000               .BYTE   1
326
327 170721 001               ;ROM TEST
328 170723 160128           .WORD   KX$T42
329 170725 001               .WORD   T42FRR
330 170727 000               .BYTE   1
331
332 170729 001               ;ROM TEST
333 170731 160132           .WORD   KX$T43
334 170733 001               .WORD   T43FRR
335 170735 000               .BYTE   1
336
337 170737 001               ;ROM TEST
338 170739 160136           .WORD   KX$T44
339 170741 001               .WORD   T44FRR
340 170743 000               .BYTE   1
341
342 170745 001               ;ROM TEST
343 170747 160140           .WORD   KX$T45
344 170749 001               .WORD   T45FRR
345 170751 000               .BYTE   1
346
347 170753 001               ;ROM TEST
348 170755 160144           .WORD   KX$T46
349 170757 001               .WORD   T46FRR
350 170759 000               .BYTE   1
351
352 170761 001               ;ROM TEST
353 170763 160148           .WORD   KX$T47
354 170765 001               .WORD   T47FRR
355 170767 000               .BYTE   1
356
357 170769 001               ;ROM TEST
358 170771 160152           .WORD   KX$T48
359 170773 001               .WORD   T48FRR
360 170775 000               .BYTE   1
361
362 170777 001               ;ROM TEST
363 170779 160156           .WORD   KX$T49
364 170781 001               .WORD   T49FRR
365 170783 000               .BYTE   1
366
367 170785 001               ;ROM TEST
368 170787 160160           .WORD   KX$T50
369 170789 001               .WORD   T50FRR
370 170791 000               .BYTE   1
371
372 170793 001               ;ROM TEST
373 170795 160164           .WORD   KX$T51
374 170797 001               .WORD   T51FRR
375 170799 000               .BYTE   1
376
377 170801 001               ;ROM TEST
378 170803 160168           .WORD   KX$T52
379 170805 001               .WORD   T52FRR
380 170807 000               .BYTE   1
381
382 170809 001               ;ROM TEST
383 170811 160172           .WORD   KX$T53
384 170813 001               .WORD   T53FRR
385 170815 000               .BYTE   1
386
387 170817 001               ;ROM TEST
388 170819 160176           .WORD   KX$T54
389 170821 001               .WORD   T54FRR
390 170823 000               .BYTE   1
391
392 170825 001               ;ROM TEST
393 170827 160180           .WORD   KX$T55
394 170829 001               .WORD   T55FRR
395 170831 000               .BYTE   1
396
397 170833 001               ;ROM TEST
398 170835 160184           .WORD   KX$T56
399 170837 001               .WORD   T56FRR
400 170839 000               .BYTE   1
401
402 170841 001               ;ROM TEST
403 170843 160188           .WORD   KX$T57
404 170845 001               .WORD   T57FRR
405 170847 000               .BYTE   1
406
407 170849 001               ;ROM TEST
408 170851 160192           .WORD   KX$T58
409 170853 001               .WORD   T58FRR
410 170855 000               .BYTE   1
411
412 170857 001               ;ROM TEST
413 170859 160196           .WORD   KX$T59
414 170861 001               .WORD   T59FRR
415 170863 000               .BYTE   1
416
417 170865 001               ;ROM TEST
418 170867 160200           .WORD   KX$T60
419 170869 001               .WORD   T60FRR
420 170871 000               .BYTE   1
421
422 170873 001               ;ROM TEST
423 170875 160204           .WORD   KX$T61
424 170877 001               .WORD   T61FRR
425 170879 000               .BYTE   1
426
427 170881 001               ;ROM TEST
428 170883 160208           .WORD   KX$T62
429 170885 001               .WORD   T62FRR
430 170887 000               .BYTE   1
431
432 170889 001               ;ROM TEST
433 170891 160212           .WORD   KX$T63
434 170893 001               .WORD   T63FRR
435 170895 000               .BYTE   1
436
437 170897 001               ;ROM TEST
438 170899 160216           .WORD   KX$T64
439 170901 001               .WORD   T64FRR
440 170903 000               .BYTE   1
441
442 170905 001               ;ROM TEST
443 170907 160220           .WORD   KX$T65
444 170909 001               .WORD   T65FRR
445 170911 000               .BYTE   1
446
447 170913 001               ;ROM TEST
448 170915 160224           .WORD   KX$T66
449 170917 001               .WORD   T66FRR
450 170919 000               .BYTE   1
451
452 170921 001               ;ROM TEST
453 170923 160228           .WORD   KX$T67
454 170925 001               .WORD   T67FRR
455 170927 000               .BYTE   1
456
457 170929 001               ;ROM TEST
458 170931 160232           .WORD   KX$T68
459 170933 001               .WORD   T68FRR
460 170935 000               .BYTE   1
461
462 170937 001               ;ROM TEST
463 170939 160236           .WORD   KX$T69
464 170941 001               .WORD   T69FRR
465 170943 000               .BYTE   1
466
467 170945 001               ;ROM TEST
468 170947 160240           .WORD   KX$T70
469 170949 001               .WORD   T70FRR
470 170951 000               .BYTE   1
471
472 170953 001               ;ROM TEST
473 170955 160244           .WORD   KX$T71
474 170957 001               .WORD   T71FRR
475 170959 000               .BYTE   1
476
477 170961 001               ;ROM TEST
478 170963 160248           .WORD   KX$T72
479 170965 001               .WORD   T72FRR
480 170967 000               .BYTE   1
481
482 170969 001               ;ROM TEST
483 170971 160252           .WORD   KX$T73
484 170973 001               .WORD   T73FRR
485 170975 000               .BYTE   1
486
487 170977 001               ;ROM TEST
488 170979 160256           .WORD   KX$T74
489 170981 001               .WORD   T74FRR
490 170983 000               .BYTE   1
491
492 170985 001               ;ROM TEST
493 170987 160260           .WORD   KX$T75
494 170989 001               .WORD   T75FRR
495 170991 000               .BYTE   1
496
497 170993 001               ;ROM TEST
498 170995 160264           .WORD   KX$T76
499 170997 001               .WORD   T76FRR
500 171000 000               .BYTE   1
501
502 171002 001               ;ROM TEST
503 171004 160268           .WORD   KX$T77
504 171006 001               .WORD   T77FRR
505 171008 000               .BYTE   1
506
507 171010 001               ;ROM TEST
508 171012 160272           .WORD   KX$T78
509 171014 001               .WORD   T78FRR
510 171016 000               .BYTE   1
511
512 171018 001               ;ROM TEST
513 171020 160276           .WORD   KX$T79
514 171022 001               .WORD   T79FRR
515 171024 000               .BYTE   1
516
517 171026 001               ;ROM TEST
518 171028 160280           .WORD   KX$T80
519 171030 001               .WORD   T80FRR
520 171032 000               .BYTE   1
521
522 171034 001               ;ROM TEST
523 171036 160284           .WORD   KX$T81
524 171038 001               .WORD   T81FRR
525 171040 000               .BYTE   1
526
527 171042 001               ;ROM TEST
528 171044 160288           .WORD   KX$T82
529 171046 001               .WORD   T82FRR
530 171048 000               .BYTE   1
531
532 171050 001               ;ROM TEST
533 171052 160292           .WORD   KX$T83
534 171054 001               .WORD   T83FRR
535 171056 000               .BYTE   1
536
537 171058 001               ;ROM TEST
538 171060 160296           .WORD   KX$T84
539 171062 001               .WORD   T84FRR
540 171064 000               .BYTE   1
541
542 171066 001               ;ROM TEST
543 171068 160300           .WORD   KX$T85
544 171070 001               .WORD   T85FRR
545 171072 000               .BYTE   1
546
547 171074 001               ;ROM TEST
548 171076 160304           .WORD   KX$T86
549 171078 001               .WORD   T86FRR
550 171080 000               .BYTE   1
551
552 171082 001               ;ROM TEST
553 171084 160308           .WORD   KX$T87
554 171086 001               .WORD   T87FRR
555 171088 000               .BYTE   1
556
557 171090 001               ;ROM TEST
558 171092 160312           .WORD   KX$T88
559 171094 001               .WORD   T88FRR
560 171096 000               .BYTE   1
561
562 171098 001               ;ROM TEST
563 171100 160316           .WORD   KX$T89
564 171102 001               .WORD   T89FRR
565 171104 000               .BYTE   1
566
567 171106 001               ;ROM TEST
568 171108 160320           .WORD   KX$T90
569 171110 001               .WORD   T90FRR
570 171112 000               .BYTE   1
571
572 171114 001               ;ROM TEST
573 171116 160324           .WORD   KX$T91
574 171118 001               .WORD   T91FRR
575 171120 000               .BYTE   1
576
577 171122 001               ;ROM TEST
578 171124 160328           .WORD   KX$T92
579 171126 001               .WORD   T92FRR
580 171128 000               .BYTE   1
581
582 171130 001               ;ROM TEST
583 171132 160332           .WORD   KX$T93
584 171134 001               .WORD   T93FRR
585 171136 000               .BYTE   1
586
587 171138 001               ;ROM TEST
588 171140 160336           .WORD   KX$T94
589 171142 001               .WORD   T94FRR
590 171144 000               .BYTE   1
591
592 171146 001               ;ROM TEST
593 171148 160340           .WORD   KX$T95
594 171150 001               .WORD   T95FRR
595 171152 000               .BYTE   1
596
597 171154 001               ;ROM TEST
598 171156 160344           .WORD   KX$T96
599 171158 001               .WORD   T96FRR
600 171160 000               .BYTE   1
601
602 171162 001               ;ROM TEST
603 171164 160348           .WORD   KX$T97
604 171166 001               .WORD   T97FRR
605 171168 000               .BYTE   1
606
607 171170 001               ;ROM TEST
608 171172 160352           .WORD   KX$T98
609 171174 001               .WORD   T98FRR
610 171176 000               .BYTE   1
611
612 171178 001               ;ROM TEST
613 171180 160356           .WORD   KX$T99
614 171182 001               .WORD   T99FRR
615 171184 000               .BYTE   1
616
617 171186 001               ;ROM TEST
618 171188 160360           .WORD   KX$T100
619 171190 001               .WORD   T100FRR
620 171192 000               .BYTE   1
621
622 171194 001               ;ROM TEST
623 171196 160364           .WORD   KX$T101
624 171198 001               .WORD   T101FRR
625 171200 000               .BYTE   1
626
627 171202 001               ;ROM TEST
628 171204 160368           .WORD   KX$T102
629 171206 001               .WORD   T102FRR
630 171208 000               .BYTE   1
631
632 171210 001               ;ROM TEST
633 171212 160372           .WORD   KX$T103
634 171214 001               .WORD   T103FRR
635 171216 000               .BYTE   1
636
637 171218 001               ;ROM TEST
638 171220 160376           .WORD   KX$T104
639 171222 001               .WORD   T104FRR
640 171224 000               .BYTE   1
641
642 171226 001               ;ROM TEST
643 171228 160380           .WORD   KX$T105
644 171230 001               .WORD   T105FRR
645 171232 000               .BYTE   1
646
647 171234 001               ;ROM TEST
648 171236 160384           .WORD   KX$T106
649 171238 001               .WORD   T106FRR
650 171240 000               .BYTE   1
651
652 171242 001               ;ROM TEST
653 171244 160388           .WORD   KX$T107
654 171246 001               .WORD   T107FRR
655 171248 000               .BYTE   1
656
657 171250 001               ;ROM TEST
658 171252 160392           .WORD   KX$T108
659 171254 001               .WORD   T108FRR
660 171256 000               .BYTE   1
661
662 171258 001               ;ROM TEST
663 171260 160396           .WORD   KX$T109
664 171262 001               .WORD   T109FRR
665 171264 000               .BYTE   1
666
667 171266 001               ;ROM TEST
668 171268 160400           .WORD   KX$T110
669 171270 001               .WORD   T110FRR
670 171272 000               .BYTE   1
671
672 171274 001               ;ROM TEST
673 171276 160404           .WORD   KX$T111
674 171278 001               .WORD   T111FRR
675 171280 000               .BYTE   1
676
677 171282 001               ;ROM TEST
678 171284 160408           .WORD   KX$T112
679 171286 001               .WORD   T112FRR
680 171288 000               .BYTE   1
681
682 171290 001               ;ROM TEST
683 171292 160412           .WORD   KX$T113
684 171294 001               .WORD   T113FRR
685 171296 000               .BYTE   1
686
687 171298 001               ;ROM TEST
688 171300 160416           .WORD   KX$T114
689 171302 001               .WORD   T114FRR
690 171304 000               .BYTE   1
691
692 171306 001               ;ROM TEST
693 171308 160420           .WORD   KX$T115
694 171310 001               .WORD   T115FRR
69
```

KXTNF - KXT11CA Native Firmware MACRO V05.00 00:51 Page 70-3
DECTST - DECODE AND DISPATCH QBUS SELF TEST REQUESTS

172 170472 160010	.WORD T12	;VECTOR ADDRESS
173	;TWO PORT RAM TEST	
174 170474 102000	.WORD KXST13	;TPR 0 VALUE
175 170476 170742	.WORD T13ERR	;LED UPDATING ROUTINE ON ERROR
176 170500 000	.BYTE 0	;NO INPUT PARAMETFR
177 170501 001	.BYTE 1	;IN ROM OVERLAY 1
178 170502 160014	.WORD T13	;VECTOR ADDRESS
179	;END OF TABLE	
180 170504 000000	.WORD 0	;IF NO MATCH, THEN COMMAND ERROR.
181		

```
1 .SBTTL TNNERR - UPDATE LEDS AFTER SELFTEST NN
2 .ENABL GBL
3
4
5 ; MODULE NAME: TNNERR - UPDATE LEDS AFTER SELFTEST NN
6 ;
7 ;
8 ;
9 ;
10 ;
11 ;
12 ; FUNCTIONAL DESCRIPTION:
13 ;
14 ; T1ERR THROUGH T13ERR HANDLE SETTING THE LEDS DEPENDING ON THE ERROR
15 ; CODE RETURNED IN R0 BY A SELF-TST ROUTINE. EACH TEST HAS A SET OF
16 ; BITS GIVING "SIGNIFICANT" ERROR CONDITIONS. "INSIGNIFICANT" ERROR
17 ; CONDITIONS ARE SUCH ERROR CODES AS "ROM MAPPED LOW SO INTERRUPT
18 ; TESTS WERE NOT EVEN TRIED". AFTER DECIDING ON AN LED VALUE, THE
19 ; ROUTINE CALLS SETLED TO SET THE VALUE. SETLED PRIORITIZES THE LED
20 ; SETTINGS SUCH THAT A NON-FATAL CODE WILL NOT REPLACE A FATAL CODE,
21 ; FOR EXAMPLE.
22 ;
23 ; INPUTS:
24 ; R0 CONTAINS THE TEST'S DISCRETE ERROR CODE.
25 ; R1 CONTAINS THE VALUE IN R1 WHEN THE SELFTEST ROUTINE WAS CALLED. THIS
26 ; VALUE IS OPTIONALY USED BY TNNERR.
27 ;
28 ; CALLING SEQUENCE:
29 ; JSR PC,TNNERR
30 ;
31 ; OUTPUTS:
32 ; THE LEDS ARE UPDATED VIA A CALL TO SETLED.
33 ;
34 ; R0 CONTAINS 0 IF NO SIGNIFICANT ERROR OCCURRED. R0 CONTAINS THE LED
35 ; VALUE USED IN THE CALL TO SETLED IF A SIGNIFICANT ERROR OCCURRED.
36 ;
37 ; NOTE: A SIGNIFICANT ERROR IS AN ERROR THAT RESULTS IN A REQUEST TO
38 ; UPDATE THE LEDS. WHEN A SELFTEST CAN NOT BE RUN (TYPICALLY BECAUSE ROM
39 ; IS MAPPED LOW), THE RESULTING ERROR IS NOT SIGNIFICANT AND, THEREFORE,
40 ; THERE IS NO REQUEST TO UPDATE THE LEDS.
41 ;
42 ; ROUTINES USED:
43 ; SETLED
44 ;
45 ; REGISTERS CHANGED:
46 ; R0,R2
47 ;
```

100

```

1 ;THE CSR TEST, TEST 1, IS A SPECIAL CASE BECAUSE WE MUST DIFFERENTIATE BETWEEN
2 ;THE FATAL NON-STANDALONE TPR ERROR AND THE OTHER NON-FATAL ERRORS.
3 170506 032700 000004      T1ERR:: BIT #4,RO      ;IS THERE A TPR ERROR?
4 170512 001407      BEQ T1ERR1      ;NO.
5 170514 032737 000340 177524      BIT #340,RWKW.CSC ;YES, BUT ARE WE IN STANDALONE MODE (I. E.
6                               ;WITH ID EQUAL TO 0 OR 1)? NOTE: THE 340 MASK
7                               ;IS A SUBSET OF THE KMSIDS DEFINITION. IF THE
8                               ;LOCATION OF KMSIDS CHANGES, THE 340 MUST BE
9                               ;CHANGED.
10 170522 001406      BEQ T1ERR2      ;YES, SO IT'S A NON-FATAL ERROR.
11 170524 012700 000017      MOV #FERRD,RO      ;NO, WE ARE IN NON-STANDALONE MODE SO IT'S A
12                               ;FATAL ERROR.
13 170530 000511      T1ERR1: BIT #373,RO      ;ARE THERE OTHER CSR ERRORS?
14 170532 032700 000373      BEQ TNNCM1      ;NO.
15 170536 001511      T1ERR2: MOV #MFERRD,RO      ;YES, SET UP THE NON-FATAL LED DISPLAY.
16 170540 012700 000010      BR TNNCM1
17 170544 000503
18
19 ;THE RAM TEST, TEST 2, IS A SPECIAL CASE BECAUSE WE MUST DIFFERENTIATE BETWEEN
20 ;FATAL NATIVE RAM ERRORS AND NON-FATAL USER RAM ERRORS.
21 170546 032700 000037      T2ERR:: BIT #37,RO      ;ARE THERE NATIVE RAM ERRORS?
22 170552 001403      BEQ T2ERR1      ;NO.
23 170554 012700 000017      MOV #MFERRD,RO      ;YES, SET UP FATAL LED DISPLAY.
24 170560 000475      BR TNNCM1
25 170562 032700 001749      T2ERR1: BIT #1740,RO      ;ARE THERE USER RAM ERRORS?
26 170566 001475      BEQ TNNINX      ;NO.
27 170570 012700 000011      MOV #UROMMOD,RO      ;YES, SET UP NON-FATAL LED DISPLAY.
28 170574 000467      BR TNNCM1
29
30 ;THE ROM TEST, TEST 3, ERROR HANDLING IS A SPECIAL CASE BECAUSE THERE ARE
31 ;3 POSSIBLE LED SETTINGS THAT MAY RESULT.
32 170576 032700 000003      T3EPR:: BIT #3,RO      ;ARE THERE NATIVE ROM ERRORS?
33 170602 001403      BEQ T3ERR1      ;NO.
34 170604 012700 000017      MOV #MFERRD,RO      ;YES.
35 170610 000461      BR TNNCM1
36 170612 032700 000004      T3ERR1: BIT #4,RO      ;IS THERE A LOW BYTE USER ROM ERROR?
37 170616 001403      BEQ T3ERR2      ;NO.
38 170620 012700 000011      MOV #UROMMOD,RO      ;YES.
39 170624 000453      BR TNNCM1
40 170626 032700 000010      T3ERR2: BIT #10,RO      ;IS THERE A HIGH BYTE USER ROM ERROR?
41 170632 001453      BEQ TNNINX      ;NO.
42 170634 012700 000012      MOV #UROMID,RO      ;YES.
43 170640 000445      BR TNNCM1
44 170642 012702 170766      T4ERR:: MOV #T4ETB,R2      ;AS ABOVE FOR THE CPU TEST.
45 170646 000437      BR TNNCOM
46 170650 012702 170772      T5ERR:: MOV #T5ETB,R2      ;AS ABOVE FOR THE BEVENT TEST.
47 170654 000434      BR TNNCOM
48 170656 012702 170776      T6ERR:: MOV #T6ETB,R2      ;AS ABOVE FOR THE DLART TEST.
49 170662 000431      BR TNNCOM
50
51 ;THE 2ND SERIAL LINE UNIT TEST, TEST 7, IS A CODED SPECIAL CASE TO HANDLE
52 ;CHANNEL A / CHANNEL B LED SETTINGS.
53 170664 005701      T7ERR:: TST R1      ;DID WE DO THE "BOTH CHANNELS GO/NO GO" TEST?
54 170666 001403      BEQ T7ERR1      ;YES, USE CHANNEL A'S LED SETTING.
55 170670 032701 000001      BIT #BIT0,R1      ;NO, DID WE SELFTEST ONLY CHANNEL A?
56 170674 001403      BEQ T7ERR2      ;NO.
57 170676 012702 171002      T7ERR1: MOV #T7AETB,R2      ;YES.

```

101

```

58 170702 000421          BR      TNNCOM
59 170704 032701 000002   T7ERR2: BIT #BIT1,R1 ;DID WE SELFTEST ONLY CHANNEL B?
60 170710 001424          BEQ    TNNINX ;NU.
61 170712 012702 171006   MOV    #T7BETB,R2 ;YES.
62 170716 000413          BR      TNNCOM
63 170720          BR      T1GERR::: ;AS ABOVE FOR THE PIO CHIP TEST.
64 170720 012702 171012   MOV    #T10ETB,R2
65 170724 000410          BR      TNNCOM
66 170726          BR      T11ERR::: ;AS ABOVE FOR THE DMA CHIP TEST.
67 170726 012702 171016   MOV    #T11ETB,R2
68 170732 000405          BR      TNNCOM
69 170734          BR      T12ERR::: ;AS ABOVE FOR THE QIR TEST.
70 170734 012702 171022   MOV    #T12ETB,R2
71 170740 000402          BR      TNNCOM
72 170742          BR      T13ERR::: ;AS ABOVE FOR THE TPR TEST.
73 170742 012702 171026   MOV    #T13ETB,R2
74
75 ;TNNCOM IS THE COMMON ENTRY TO USE WHEN R2 POINTS TO A TABLE ENTRY OF
76 ;SIGNIFICANT ERROR BITS FOLLOWED BY LED VALUE.
77 170746 032200          TNNCOM: BIT (R2)+,R0 ;ARE ANY OF THE "SIGNIFICANT" ERROR BITS
78                      BEQ    TNNINX ;SET IN THE RETURNED ERROR CODE?
79 170750 001404          MOV    (P2),R0 ;NO, DON'T EVEN TRY TO UPDATE THE LEDS.
80 170752 011200          BR      TNNCM1
81
82 ;TNNCM1 IS THE COMMON ENTRY TO USE WHEN R0 CONTAINS AN LED SETTING.
83 170754 004737 167200   TNNCM1: JSR    PC,@RESETLED ;UPDATE THE LEDS USING SETLFD SO THAT A MORE
84                      ;IMPORTANT SETTING DOES NOT GET OVERWRITTEN.
85 170760 000401          BR      TNNXIT
86
87 ;TNNINX IS THE EXIT WHEN THERE ARE NO SIGNIFICANT ERRORS.
88 170762 005000          TNNINX: CLR    R0 ;SHOW THAT NO SIGNIFICANT ERROR OCCURRED.
89 170764 000207          TNNXIT: RTS   PC
90
91 ;ERROR TABLES GIVING THE "SIGNIFICANT" ERROR BIT SETTINGS AND THE ASSOCIATED
92 ;LED SETTING FOR THE ERROR. THE SPARSNESS OF THE BITS IS BECAUSE CERTAIN
93 ;TESTS RETURN NON-ZERO ERROR CODES THAT REALLY MEAN "COULDN'T RUN THE TEST".
94 ;THERE ARE SEVERAL TESTS WHERE INTERRUPTS CANNOT BE TESTED BECAUSE ROM IS IN
95 ;LOW MEMORY.
96
97 ;T1, T2, & T3 MUST BE SPECIAL CASED IN CODE.
98 170766 007777 000017   T4ETB: .WORD 7777,FERRD
99 170772 000016 000010   T5ETB: .WORD 16,NFFRPD
100 170776 000176 000013  T6ETB: .WORD 176,S1LRD
101 171002 007476 000014  T7AETR: .WORD 7476,S2ALBD ;NOTE: BITS 6 & 7 ARE NOT INCLUDED BECAUSE A
102
103
104
105
106
107
108 171006 007776 000015  T7BETR: .WORD 7776,S2RLRD ;NOTE: DMA (AND THEREFORE BITS 6 & 7) DOES
109
110
111
112 171012 000376 000016  T10ETB: .WORD 376,PAPLBD ;NOT APPLY TO CHANNEL B. IF THE BITS COME ON
113 171016 000074 000010   T11ETB: .WORD 74,NFFPPD ;WE HAVE A SERIOUS ERROR, SO WE OUGHT TO SET
114 171022 000172 000010   T12ETB: .WORD 172,NFERRD ;THE LEDS.
115 171026 007767 000017   T13ETB: .WORD 7767,FERRD
116

```

102

```
1          .SATTL D4ALD - DMA LOADER
2          .ENABLE LC,GRU
3
4          ;
5          ; Module name: D4ALD - DMA LOAD
6          ;
7          ; System: KXT11-CA Native Firmware
8          ;
9          ;
10         ;
11         ;
12         ; Functional Description:
13         ;
14         ; This is the module that controls the loading of the IOP's application
15         ; from the Q bus by the DMA controller. It is used by the Q bus command
16         ; interpreter module on reception of a DMA load command. This module
17         ; expects the data which is to be placed into the chain control registers
18         ; of the DMA controller to be in the dual port registers 2 and 3.
19         ; On completion of the load, this module places the IOP in the
20         ; Wait state.
21
22         ;Input Parameters:
23
24         ;      NONE
25
26         ;Output Parameters
27
28         ;      NONE
29
30         ;DATA STRUCTURES USED
31
32         ;      1) DMA MASTER MODE REGISTER
33         ;      2) DMA COMMAND REGISTER
34         ;      3) DMA CHANNEL 0 REGISTER
35         ;      4) DUAL PORT RAM REGISTER 2 AND 3
36         ;      5) IOP STATUS REGISTER
37         ;      6) IOP COMMAND REGISTER
38
39         ;Routines Used
40
41         ;      1) WAITST (JUMP TO)
42
43         ;Global Symbols Defined
44
```

1	174442	CHNOFO	=	174442	;DMA CHAIN OFFSET REGISTER ADDRESS OF CHANNEL 0
2	174446	CHNST0	=	174446	;DMA CHAIN SEGMENT/TAG REGISTER ADDRESS OF CHANNEL 0
3	174470	MMREGA	=	174470	;DMA MASTER MODE REGISTER ADDRESS
4	174454	CMREGA	=	174454	;DMA COMMAND REGISTER ADDRESS
5	174456	SOREGA	=	174456	;DMA CHANNEL 0 STATUS REGISTER ADDRESS
6	010000	CAEIT	=	10000	;CHAIN ABORT BIT OF STATUS
7	000002	EOPBIT	=	2	;END OF PROCESS BIT OF STATUS
8	004000	NACRIT	=	4000	;NAC BIT OF STATUS

```

1                               ;* PROCEDURE DMA LOAD
2 171032                      DMA LD:::                         ;* * SET LEDs TO DMA DISPLAY
3
4 171032 012700 000002          MOV   #DMAED,R0
5 171036 012701 171046          MOV   #DMA50,R1
6 171042 000167 176134          JMP   SETLDJ
7 171046
8
9 171046 112737 000151 174470  MOVB  #151, @#MMREGA      ;* * SET DMA MASTER MODE REGISTER TO NO INTERRUPT
10
11 171054 013737 175004 174442  MOV   @#KW.SC2, @#CHNOFO
12 171062 013737 175006 174446  MOV   @#KW.SC3, @#CHNSTO
13
14 171070 112737 000202 174454  MOVB  #202, @#CMREGA     ;* * SET HARDWARE MASK
15
16 171076 112737 000240 174454  MOVB  #240, @#CMREGA     ;* * START CHAIN LOAD CHANNEL 0
17
18 171104
19
20 171104 032737 014002 174456  BIT   #CCABIT1, EOPBIT1, NACBIT1, @#SOREGA
21 171112 001774               BEQ   DMA100
22
23 171114
24
25 171114 013737 174456 175004  DMA200:             ;* * COPY DMA STATUS REGISTER CHANNEL 0
26
27
28 171122 032737 010002 174456  MOV   @#SOREGA, @#KW.SC2
29 171130 001004               BIT   #CCABIT1, EOPBIT1, @#SOREGA
30
31
32 171132 042737 040000 175002  DMA300:             ;* * IF NORMAL COMPLETION
33 171140 000403               BIC   #KXSDME, @#KW.STA
34
35 171142
36
37 171142 052737 140000 175002  DMA400:             ;* * CLEAR DMA ERROR BIT OF DPR
38
39 171150
40
41 171150 012700 171160  DMA500:             ;* * END IF
42
43 171150 000167 002526  MOV   #DMA600,R0
44 171154 000167 002526  JMP   SPR5SU
45 171160
46
47 171160 005037 175000  DMA600:             ;* * SETUP STACK POINTER AND R5
48
49 171164 000167 176212  CLR   @#KW.CMD
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
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
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
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
789
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
889
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
917
918
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
988
989
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1016
1017
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1037
1037
1038
1039
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1059
1060
1061
1062
1063
1064
1065
1066
1067
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1089
1090
1091
1092
1093
1094
1095
1096
1097
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1109
1110
1111
1112
1113
1114
1115
1115
1116
1117
1118
1119
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1159
1160
1161
1162
1163
1164
1165
1166
1167
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1189
1190
1191
1192
1193
1194
1195
1196
1197
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1209
1210
1211
1212
1213
1214
1215
1216
1217
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1289
1290
1291
1292
1293
1294
1295
1296
1297
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1309
1310
1311
1312
1313
1314
1315
1316
1317
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1389
1390
1391
1392
1393
1394
1395
1396
1397
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1409
1410
1411
1412
1413
1414
1415
1416
1417
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1489
1490
1491
1492
1493
1494
1495
1496
1497
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1509
1510
1511
1512
1513
1514
1515
1516
1517
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1589
1590
1591
1592
1593
1594
1595
1596
1597
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1609
1610
1611
1612
1613
1614
1615
1616
1617
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1689
1690
1691
1692
1693
1694
1695
1696
1697
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1709
1710
1711
1712
1713
1714
1715
1716
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1789
1790
1791
1792
1793
1794
1795
1796
1797
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1889
1890
1891
1892
1893
1894
1895
1896
1897
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1989
1990
1991
1992
1993
1994
1995
1996
1997
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2089
2090
2091
2092
2093
2094
2095
2096
2097
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2149
2150
2151
2152
215
```

1 .SBTTL BSUPV - BOOT SUPERVISOR
2 .ENABLE LC,GBL
3
4
5 ;
6 ; Module name: BSUPV - ROOT SUPERVISOR
7 ;
8 ; System: KXT11-CA Native Firmware
9 ;
10 ;
11 ;
12 ;
13 ;
14 ;
15 ; Functional Description:
16 ;
17 ; This module supervises the boot process. It makes the determination
18 ; of which device the application will be loaded from and transfers
19 ; control to the appropriate routines.
20
21 ;Input Parameters:
22 ;
23 ; 1. BOOTFD of FLGWRD
24 ;
25 ;
26 ;Output Parameters
27 ;
28 ; NONE
29 ;
30 ;Data Structures Used
31 ;
32 ; 1. LED DISPLAY
33 ;
34 ; 2. IOP STATUS REGISTER
35 ;
36 ; 3. TCSR8, bits of BOOTSW
37 ;
38 ;Routines Used
39 ;
40 ; 1. TU58 Boot (TU58BT)
41 ;
42 ; 2. Loop back tests (LBTEST)
43 ;
44 ; 3. Serial ODT monitor (SODTM)
45 ;
46 ; 4. Set LEDs (SETLED)
47 ;
48 ; 5. Trap 24 emulator (TRAP24)
49 ;
50 ;Global Symbols Defined
51 ;
52 ; NONE

105

```

1          ;* PROCEDURE BOOTSTRAP SUPERVISOR
2 171170          BSUPV::                                ;** * SET SP TO DEFAULT USER STACK
3
4 171170 010500          MOV    R5,R0
5 171172 062700 177700          ADD    #ADSTKO,R0
6 171176 010006          MOV    R0,SP
7
8
9
10
11 171200 022765 000002 177774          CMP    #2,BOOTFO(R5)
12 171206 103414          BLO    BS100
13
14 171210 012700 000000          MOV    #NNCED,R0
15 171214 004767 175760          CALL   SETLFD
16
17
18 171220 042737 000007 175002          BIC    #KMSSTF,@#KW.STA
19 171226 052737 000007 175002          BIS    #KPSNNC,@#KW.STA
20
21 171234 000167 176116          JMP    TRAP24
22
23 171240 022765 000003 177774  BS100:  CMP    #3,BOOTFO(R5)
24 171246 001014          BNE    BS200
25
26 171250 012700 000001          MOV    #PRIAD,R0
27 171254 004767 175720          CALL   SETLED
28
29 171260 042737 000007 175002          BIC    #KMSSTF,@#KW.STA
30 171266 052737 000005 175002          BIS    #KPSPRS,@#KW.STA
31
32 171274 000167 000434          JMP    TU58BT
33
34 171300 022765 000004 177774  BS200:  CMP    #4,BOOTFO(R5)
35 171306 001007          BNE    BS300
36
37 171310 052737 020000 175002          BIS    #KXSBDF,@#KW.STA
38
39 171316 005046          CLR    -(SP)
40 171320 005046          CLR    -(SP)
41
42 171322 000167 173300          JMP    SODTM ;(NON-STRUCTURED EXIT TO SERIAL ODT)
43
44 171326 022765 000005 177774  BS300:  CMP    #5,BOOTFO(R5)
45 171334 001404          BEQ    BS400
46
47 171336 022765 000006 177774          CMP    #6,BOOTFO(R5)
48 171344 001002          BNE    BS500
49 171346          BS400:
50
51 171346 000167 176030          JMP    WAITST ;(NON STRUCTURED EXIT)
52
53 171352          BS500:                                ;** * WHEN 8
54 171352 022765 000010 177774          CMP    #8.,BOOTFO(R5)
55 171360 001404          BF4    BS550
56
57 171362 022765 000011 177774          CMP    #9.,BOOTFO(R5)

```

KXTNF - KXT11CA Native Firmware MACRO V05.00 00:51 Page 77-1
BSUPV - BOOT SUPERVISOR

58 171370 001005		BNE	BS570	
59 171372		BS550:		
60				;*** ENABLE OVERLAY 0
61 171372 042737 000040 177520		BIC	#KXSDPE, #KKW.CSA	;*** PERFORM LOOP BACK TESTS
62		JMP	LRTST ;(NON-STRUCTURED EXIT)	;*** WHEN 10
63 171400 000167 172306				
64		BS570:		
65 171404		CMP	#10., #BOOTFO(R5)	
66 171404 022765 000012 177774		BNE	BS600	
67 171412 001002				;*** WAIT FOR COMMAND
68		JMP	WAITST ;(NON-STRUCTURED EXIT)	;*** ELSE
69 171414 000167 175762				
70		BS600:		
71 171420		MOV	#ERRRD, R0	;*** SET LEDS TO FATAL ERROR
72		CALL	SETLED	
73 171420 012700 000017				;*** INVALID SELECTION STOP IOP
74 171424 004767 175550		FAS::	BR	
75				;*** END CASE
76 171430 000777				;*** END BOOTSTRAP SUPERVISOR
77				
78				

```
1 .SBTTL QODTM - Q BUS ODT MONITOR
2
3 .FENABLE GBL,LC
4 ;
5 ; Module name: QODTM - Q BUS ODT MONITOR
6 ;
7 ; System: KXT11-CA Native Firmware
8 ;
9 ;
10 ;
11 ;
12 ; Functional Description:
13 ;
14 ; This module is the main controlling module for the Q bus ODT mode.
15 ; It supervises the ODT functions and on completion returns control
16 ; back to the application or places the IOP in the wait state.
17 ;
18 ;Input Parameters:
19 ;
20 ; None
21 ;
22 ;Output Parameters
23 ;
24 ; None
25 ;
26 ;Routines Used
27 ;
28 ; 1) Save Registers (SAVERG)
29 ; 2) Set LEDs (SETLED)
30 ; 3) Wait State (WAITST)
31 ; 4) Go/Proceed (PROC)
32 ; 5) Go/Proceed (GO)
33 ; 6) Deposit (DEPOS)
34 ; 7) Examine (EXAMIN)
35 ;
36 ;Registers Used Outside of Output Parameters:
37
38
```

108

```

1 171432                                QODTM::: ;* PROCEDURE Q BUS ODT MONITOR
2
3
4
5 171432 052767 010200 003342          BIS    #<KXSNXF ! KXSQOF>,KW STA ;* * SET ODT FORMAT FLAG OF STATUS REGISTER
6
7
8 171440 000167 173006          JMP    SAVERG ;* * SET NXM FLAG OF STATUS REGISTER
9 171444                                QSRET::: ;* * SAVE REGISTERS (NON STRUCTURED JUMP TO
10
11 171444 042737 000007 175002          BIC    #KMSSTF,@#KW STA ;* * SET STATUS REGISTER TO Q ODT MODE STATE AND
12 171452 052737 000003 175002          BIS    #KPSQOD,@#KW STA ;* * SET LEDS TO Q BUS ODT STATE
13
14 171460 012700 000005          MOV    #QUDTMD,RO ;* * SET COMMAND REGISTER
15 171464 004767 175510          CALL   SFTLED ;* * CLEAR ALL FLAGS
16
17 171470 005003          CLR    R3 ;* * CLEAR COMMAND REGISTER
18
19 171472 005037 175000          CLR    @#KW.CMD ;* * DO UNTIL RECEIVE EXIT COMMAND
20
21 171476                                QS100: ;* * * DO UNTIL COMMAND REGISTER NOT EMPTY
22
23
24
25 171476                                QS200: ;* * * END UNTIL
26
27 171476 000001          WAIT   ;* * * CLEAR COMMAND ERROR BIT
28
29 171500 042737 100000 175002          BIC    #KXSCME,@#KW STA ;* * CASE COMMAND RECEIVED
30
31 171506 013700 175000          MOV    @#KW.CMD,RO ;* * WHEN EXIT
32
33 171512 022700 100000          CMP    #KXSEXO,RO ;* * * * CLEAR QBUS ODT FLAG OF STATUS REG.
34 171516 001007          BNE    QS300 ;* * * * CLEAR COMMAND REGISTER
35
36 171520 042737 000200 175002          BIC    #KX$QOF,@#KW STA ;* * * ENTER WAIT STATE
37
38 171526 005037 175000          CLR    @#KW.CMD ;* * * WHEN PROCEED
39
40 171532 000167 175644          JMP    WAITST ;(NON STRUCTURED EXIT)
41
42 171536                                QS300: ;* * * EXECUTE PROCEED
43 171536 022700 000020          CMP    #KXSPRG,RO ;* * * IF ERROR (RETURN FROM PROC)
44 171542 001006          BNE    QS400 ;* * * THEN
45
46 171544 004767 175240          CALL   PPOC ;* * * * SET COMMAND ERROR BIT OF STATUS REG.
47
48
49
50 171550 052737 100000 175002          BIS    #KXSCME,@#KW STA ;* * * END IF
51
52 171556 000454          BR    QS1000 ;* * * WHEN GO
53
54 171560                                QS400: ;* * * SET GO ADDRESS = DPR 3
55 171560 022700 000010          CMP    #KX$GO0,RO
56 171564 001012          BNE    QS500
57

```

KXTNF - KXT11CA Native Firmware MACRO V05.00 00:51 Page 79-1
 QODTM - Q BUS ODT MONITOR

```

58 171566 013700 175006      MOV    @#KW.SC3,RC      ;* * * * PERFORM GU
59
60 171572 004767 175174      CALL   GO
61
62
63
64 171576 052703 100000      BTS    #ERRFCB,R3      ;* * * * IF RETURNED
65
66
67 171602 052737 000100 177530     BTS   #KXSDEN,@#KW.CSD
68 171610 000437             BR    QS1000      ;* * * * THEN
69
70 171612             QS500:      ;* * * * SET ERROR FLAG
71 171612 022700 000004      CMP    #KXSDE0,RO
72 171616 001005             BNE   QS700      ;* * * * RE-ENABLE DUAL PORT RAM (DISABLED
73
74 171620 013704 175004      MOV    @#KW.SC2,R4
75
76 171624 004767 174644      CALL   DEPOS      ;* * * * PERFORM DEPOSIT
77 171630 000427             BR    QS1000      ;* * * WHEN DEPOSIT
78
79 171632             QS700:      ;* * * WHEN OPEN REGISTER
80 171632 022700 000002      CMP    #KXSORD,RO
81 171636 001007             BNE   QS800      ;* * * * SET DATA = DPR 2
82
83 171640 013702 175006      MOV    @#KW.SC3,R2
84
85 171644 052703 000200      BTS    #REGFCB,R3      ;* * * * SET REG/ADDR FLAG = REGISTER
86
87 171650 004767 174066      CALL   EXAMIN      ;* * * * PERFORM OPEN ON REGISTER
88 171654 000415             BR    QS1000      ;* * * WHEN OPEN MEMORY
89
90 171656             QS800:      ;* * * WHEN OPEN MEMORY
91 171656 022700 000001      CMP    #KXSOM0,RO
92 171662 001007             BNE   QS900      ;* * * * SET TARGET = DPR 3
93
94 171664 013702 175006      MOV    @#KW.SC3,R2
95
96 171670 042703 000200      BIC    #REGFCB,R3      ;* * * * SET REG/ADDR FLAG = ADDRESS
97
98 171674 004767 174042      CALL   EXAMIN      ;* * * * PERFORM OPEN MEMORY
99 171700 000403             BR    QS1000      ;* * * ELSE
100
101 171702             QS900:      ;* * * * SET COMMAND ERROR BIT
102
103 171702 052737 100000 175002     BTS   #KXSCME,@#KW.STA
104
105 171710             QS1000:      ;* * * END CASE
106
107 171710 005703             TST    R3
108 171712 100005             JPL    QS1100      ;* * * * IF &ERROR FLAG SET
109
110
111 171714 052737 100000 175002     BTS   #KXSCME,@#KW.STA      ;* * * * THEN
112
113 171722 042703 100000             BIC   #ERRFCB,R3      ;* * * * SET COMMAND ERROR FLAG OF STATUS REGISTER
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
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
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
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
687
688
689
689
690
691
692
693
694
695
696
697
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
718
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
788
789
789
790
791
792
793
794
795
796
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
888
889
889
890
891
892
893
894
895
896
897
898
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
988
988
989
989
990
991
992
993
994
995
996
997
998
999
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1098
1099
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1188
1189
1189
1190
1191
1192
1193
1194
1195
1196
1197
1197
1198
1199
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1288
1289
1289
1290
1291
1292
1293
1294
1295
1296
1297
1297
1298
1299
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1389
1390
1391
1392
1393
1394
1395
1396
1397
1397
1398
1399
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1488
1489
1489
1490
1491
1492
1493
1494
1495
1496
1497
1497
1498
1499
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1588
1589
1589
1590
1591
1592
1593
1594
1595
1596
1597
1597
1598
1599
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1688
1689
1689
1690
1691
1692
1693
1694
1695
1696
1697
1697
1698
1699
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1788
1789
1789
1790
1791
1792
1793
1794
1795
1796
1797
1797
1798
1799
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1898
1899
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1988
1989
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1998
1999
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2098
2099
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
```

KXTNF - KXT11CA Native Firmware MACRO V05.00 00:51 Page 79-2
QODTM - Q BUS ODT MONITOR

```
115 171726 QS1100:  
116 ;* * * CLEAR COMMAND REGISTER  
117 171726 005037 175000 CLR #KW.CMD  
118 ;* * END UNTIL  
119 171732 000661 BR QS100  
120 ;* END Q BUS ODT MONITOR
```

OII

KXTNF - KXT11CA Native Firmware MACRO V05.00 00:51 Page 80
TU58BT - TU58 BOOT STRAP

```
1 .SBTTL TU58BT - TU58 BOOT STRAP  
2  
3 .ENABLE GBL,LC  
4 ;  
5 ; Module name: LJD  
6 ;  
7 ; System: KXT11-CA Native Firmware  
8 ;  
9 ;  
10 ;  
11 ;  
12 ;  
13 ;  
14 ; Functional Description:  
15 ;  
16 ; This is the primary loader for TU58s. It will attempt to load and transfer  
17 ; control to the code in the boot block of a TU58 drive connected to the  
18 ; console port (SLU1). If the first word of the block is in the range of  
19 ; 240 to 277 it is considered a valid boot block. This routine will  
20 ; continue to try to boot until it succeeds.
```

1
2
3 171734 TU58BT:: ;* PROCEDURE TU58 BOOT
4
5 171734 012700 000001 MOV #1,R0 ;* * SET DRIVE FLAG TO 0
6
7 171740 TU100: ;* * DO UNTIL SUCCESSFUL BOOT
8
9 171740 005100 COM R0 ;* * * TOGGLE DRIVE FLAG
10 171742 042700 177776 BIC #^C1,R0
11
12 171746 005004 CLR R4 ;* * * SET POINTER = 0
13
14 171750 004767 000260 CALL SNDBRK ;* * * SEND BREAK
15
16 171754 012702 000002 MOV #2,R2 ;* * * SEND INIT
17 171760 012701 172320 MOV #INITST,R1
18 171764 004767 000310 CALL SEND ;* * * DO UNTIL TIMED OUT OR CONTINUE RECEIVED
19
20 171770 012703 000002 MOV #2,R3
21 171774 TU120: ;* * END UNTIL
22 171774 005002 CLR R2
23 171776 TU125: ;* * * SEND BREAK
24 171776 105737 177560 TSTB @#RCSR1A
25 172002 100403 BMI TU130
26 172004 077204 SOB R2,TU125
27 172006 077306 SOB R3,TU120
28
29 172010 000753 BR TU100 ;NON STRUCTURED BRANCH
30 172012 TU130: ;* * * SEND BOOT COMMAND
31 172012 122737 000020 177562 CMPB #20,@#RBUF1A
32 172020 001365 BNE TU120
33
34
35 172022 004767 000206 CALL SNDBRK ;* * * SEND BREAK
36
37 172026 012702 000002 MOV #2,R2 ;* * * SEND BOOT COMMAND
38 172032 012701 172322 MOV #BOOTST,R1
39 172036 004767 000236 CALL SEND ;* * * DO UNTIL TRANSMIT BUFFER EMPTY
40
41 172042 TU150: ;* * * END UNTIL
42 172042 105737 177564 TSTB @#XCSR1A
43 172046 100375 BPL TU150 ;* * * SEND DRIVE NUMBER
44
45
46 172050 110067 005512 MOVB R0,XBUF1A ;* * * DO UNTIL 512 BYTES RECEIVED
47
48 172054 012702 001000 MOV #512,,R2
49 172060 TU200: ;* * * OR TIMED OUT
50
51 172060 012703 000100 MOV #100,R3
52 172064 005001 TU300: CLR R1
53 172066 105737 177560 TU400: TSTB @#RCSR1A
54 172072 100403 BMI TU500
55 172074 077104 SOB R1,TU400
56 172076 077306 SOB R3,TU300 ;* * * IF TIMED OUT
57

112

```

58 ;* * * * THEN
59 ;* * * * RETRY
60 172100 000717 BR TU100 ;(NON STRUCTURED BRANCH)
61 ;* * * * ELSE
62 172102 TU500:
63 MOVB @4KBUFLA,(R4)+ ;* * * * INPUT DATA
64 172102 113724 177562 DEC R2 ;* * * * DECREMENT BYTE COUNTER
65
66 172106 005302
67
68 172110 TU600:
69
70 172110 022702 000775 CMP #509,,R2 ;* * * * IF 3 BYTES HAVE BEEN RECEIVED
71 172114 001013 BNE TU600 ;* * * * AND BYTE 1 = 2
72
73 172116 005001 CLR R1
74 172120 122721 000002 CMPB #2,(R1)+
75 172124 001007 BNE TUB00 ;* * * * AND BYTE 2 = 12
76
77 172126 122721 000012 CMPR #12,(P1)+
78 172132 001004 BNE TU800 ;* * * * AND BYTE 3 = 100
79
80 172134 122711 000100 CMPR #100,(R1)
81 172140 001001 BNE TU800 ;* * * * THEN
82 ;* * * * RETRY
83
84 172142 000676 BR TU100 ;* * * * END IF
85
86 172144 TU800:
87
88 172144 TU900:
89 172144 005702 TST R2
90 172146 001344 BNE TU200 ;* * * END UNTIL
91
92 172150 TU100:
93 172150 005002 CLR R2
94 172152 021227 000240 CMP (R2),#240
95 172156 103670 BLO TU100
96 172160 021227 000277 CMP (R2),#277
97 172164 101265 BHI TU100 ;* * SAVE UNIT NO.
98
99 172166 010004 MOV R0,R4 ;* * SET LED DISPLAY TO NON NATIVE CODE EXECUTING
100
101 172170 012700 000000 MOV #NNCED,R0
102 172174 004767 175000 CALL SETLED ;* * SET STATUS REGISTER TO NON NATIVE CODE STATE
103
104 172200 042737 000007 175002 BIC #KHSSTF,@#KW.STA
105 172206 052737 000007 175002 BTS #KPSNNC,@#KW.STA ;* * SET STACK POINTER TO DEFAULT USER STACK
106
107 172214 012706 177700 MOV #ADSTK0,SP
108 172220 060506 ADD R5,SP ;* * PASS CSR ADDRESS TO SECONDARY LOADER
109
110 172222 012701 177560 MOV #RCSK1A,R1 ;* * PASS UNIT NUMBER TO SECONDARY LOADER
111
112 172226 010400 MOV R4,R0 ;* * EXECUTE AT 0
113
114 172230 000137 000000 JMP @#0

```

KXTNF - KXT11CA Native Firmware MACRO V05.00 00:51 Page 81-2
TU58BT - TU58 BOOT STRAP

115

;* END TU58 BOOT

KXTNF - KXT11CA Native Firmware MACRO V05.00 00:51 Page 82
TU58BT - TU58 BOOT STRAP

113

```

1 172234 SNDBRK: ;* PROCEDURE BREAK
2 172234 012702 000010 MOV #8.,R2
3 172240 105737 177564 SN100: TSTB @#XCSR1A ;* * SEND BREAK FOR 8 CHAR. TIMES
4 172244 100375 BPL SN10C
5 172246 052737 000001 177564 BIS #1,@#XCSR1A
6 172254 005037 177566 CLR @#XBUF1A
7 172260 077211 SOB R2,SN100
8
9
10
11
12 172262 105737 177564 SN200: TSTB @#XCSR1A ;* * END UNTIL
13 172266 100375 BPL SN200 ;* * DO UNTIL TRANSMIT BUFFER EMPTY
14
15
16 172270 042737 000001 177564 BIC #1,@#XCSR1A ;* * END UNTIL
17
18 172276 000207 RETURN ;* * CLEAR BREAK
19
20
21
22
23
24
25
26
27
28 172300 SEND: ;* * RETURN FROM CALL
29
30
31 172300 SD100: ;* END BREAK
32 172300 105737 177564 TSTB @#XSP1A ;* PROCEDURE SEND
33 172304 100375 BPL SD100 ;* * DO UNTIL DATA COUNT = 0
34
35
36 172306 112137 177566 MOVP (R1)+,@#XBUF1A ;* * * DO UNTIL TRANSMIT BUFFER EMPTY
37
38 172312 005302 DFC R2 ;* * * END UNTIL
39
40 172314 001371 ANE SD100 ;* * DECREMENT DATA COUNTER
41
42 172316 000207 RETURN ;* * END UNTIL
43
44
45

```

KXTNF - KXT11CA Native Firmware MACRO V05.00 00:51 Page 83
TU58BT - TU58 BOOT STRAP

```
1 172320    004           INITST: .BYTE 4
2 172321    004           .BYTE 4
3
4 172322    004           BOOTST: .BYTE 4
5 172323    C10          .BYTE 10
6
7
8
```

KXTNF - KXT11CA Native Firmware MACRO V05.00 00:51 Page 84
PUTEST - POWER UP SELF TEST

```
1
2
3
4
5 ; SBTTL PUTEST - POWER UP SELF TEST
6 ; .ENABLE GBL,LC
7
8
9
10
11
12 ; Functional Description:
13
14 ; This module controls the execution of the Power Up Auto Self Tests.
15 ; It determines which of the test are to be executed per the Boot/Self Test
16 ; selection. This routine also tests the Status Register of the DPR regardless
17 ; of the Boot/Self Test Selection and halts the KXT if the test fails.
18
19 ; Note:
20
21 ; There are two important addresses with regard to the Power Up Self
22 ; Tests. FARSTR (defined in this module) is the address the KXT will be
23 ; looping at if the Status Register is found to be faulty. FTLST (defined
24 ; in the EXTEST module) is the address that the KXT will be looping at if
25 ; any other fatal self test error is found.
26
27
28 ; Input Parameters:
29
30 ;     BOOTFO(R5) - Boot selection location
31
32 ; Output Parameters:
33
34 ;     None
35
36 ; Routines Used:
37
38 ;     1) Set LEDs (SETLFD)
39 ;         2) Execute Test (EXTEST)
40
41
42
```

```

1 172324          PUTEST::                                ;* PROCEDURE POWER UP SELF TEST
2                                         CLR      #KKW.SC2   ;* * CLEAR ALL ERROR BITS IN DPR 2
3
4 172324 005037 175004
5                                         CLR      #KKW.SC2
6
7                                         MOV      #KKW.STA,R2
8 172330 012702 175002   MOV      (R2),R0
9 172334 011200
10 172336 012701 052525  MOV      #52525,P1
11 172342 010112  MOV      R1,(R2)
12 172344 020112  CMP      R1,(R2)
13 172346 001004  BNE      PU20
14 172350 005112  COM      (R2)
15 172352 005101  COM      R1
16 172354 020112  CMP      R1,(R2)
17 172356 001405  BEQ      PU30
18
19 172360          PU20:                                ;* * IF TEST FAILED
20
21
22 172360 012700 000017  MOV      #FERRD,R0
23 172364 004767 174610  CALL    SETLED
24
25 172370 000777  FARSTR:: BR
26
27 172372          PU30:                                ;* * THEN
28                                         MOV      R0,(R2)   ;* * SET LEDS AND STATUS REGISTER TO FATAL ERROR
29 172372 010012
30                                         MOV      R0,(R2)
31 172374 016500 177774  MOV      BOOTFO(R5),R0
32 172400 005700  TST      R0
33 172402 001414  BEQ      PU50
34 172404 022700 000004  CMP      #4,R0
35 172410 101013  BHI      PU100
36                                         CMP      #5,R0   ;* * OR 5
37 172412 022700 000005  BEQ      PU100
38 172416 001410
39                                         CMP      #8,R0   ;* * OR 8
40 172420 022700 000010  BEQ      PU100
41 172424 001405
42                                         CMP      #10,R0  ;* * OR 10
43 172426 022700 000012  BEQ      PU100
44 172432 001402
45 172434 000167 000126  PU50:    JMP      PU1010
46 172440          PU100:                                ;* * THEN
47                                         BTC      #^CKMSSTP>,(R2) ;* * SET STATUS REGISTER TO AUTO TEST MODE STATE
48                                         BIS      #KPSSTS,(R2)
49 172440 042712 177770
50 172444 052712 000001  CALL    #PUTSDS,R0
51                                         SETLED
52 172450 012700 000007  MOV      #PUTSDS,R0
53 172454 004767 174520  CALL    #PUTSDS,R0
54                                         SETLED
55 172460 052767 000100 005042  BIS      #KXSDEN,KW.CSF ;* * ENABLE DPR
56                                         MOV      #4,R2   ;* * * TEST CPU
57 172466 012702 000004

```

KXTNF - KXT11CA Native Firmware MACRO V05.00 00:51 Page 85-1
 PUTEST - POWER UP SELF TEST

```

58 172472 004767 000072          CALL    EXTEST
59                                         ;* * * TEST RAM
60 172476 012702 000002          MOV     #2,R2
61 172502 004767 000062          CALL    EXTEST
62                                         ;* * * IF BOOT/SELF TEST SELECTION = 2
63 172506 022765 000002 177774      CMP    #2,BOOTFO(R5)
64 172514 001404                 BFQ    PU400
65                                         ;* * * OR 9
66 172516 022765 000011 177774      CMP    #9,,BOOTFO(R5)
67 172524 001003                 BNE    PU500
68 172526                               PU400:
69                                         ;* * * ThEN
70                                         ;* * * INCLUDE USER ROM IN TEST
71 172526 012701 000001          MOV     #1,R1
72 172532 000401                 BR    PU600
73                                         ;* * * FALSE
74 172534                               PU500:
75                                         ;* * * DO NOT INCLUDE USER ROM IN TEST
76 172534 005001                 CLR    R1
77                                         ;* * * FND IF
78 172536                               PU600:
79                                         ;* * * TEST ROM
80 172536 012702 000003          MOV     #3,R2
81 172542 004767 000022          CALL    EXTEST
82                                         ;* * * TEST CSR
83 172546 012702 000001          MOV     #1,R2
84 172552 004767 000012          CALL    EXTEST
85                                         ;* * * TEST LOCAL DMA
86 172556 012702 000011          MOV     #11,R2
87 172562 004767 000002          CALL    EXTEST
88                                         ;* * END IF
89 172566                               PU1010:
90                                         ;* * RETURN FROM CALL
91 172566 000207                 RETURN
92                                         ;* END POWER UP SELF TEST

```

KXTNF - KXT11CA Native Firmware MACRO V05.00 00:51 Page 86
 EXTEST - EXECUTE TEST

```

1           .SBTTL EXTEST - EXECUTE TEST
2
3           .ENABLE GBL,LC
4
5           ; Module name: EXTEST - EXECUTE TEST
6
7           ; System: KXT11-CA Native Firmware
8
9
10
11
12           ; Functional Description:
13
14           ; This module is used by PUTEST to call a particular self test and display any
15           ; error results in the LEDs and the DPR. It will also halt the KXT if a fatal
16           ; error is found.
17
18           ; Input Parameters:
19
20           ;     R1 = TEST PARAMETERS
21           ;     R2 = TEST NUMBER

```

```

1          ;* PROCEDURE TEST
2 172570      EXTEST:::           ;* * INITIALIZE IO
3
4 172570 004767 000616      CALL   DINIT
5
6 172574 010246      MOV    R2,-(SP)    ;* * SAVE TEST NUMBER
7
8 172576 013746 177524      MOV    @#KW.CSC,-(SP)  ;* * SAVE LED STATE
9
10 172602 010146      MOV   R1,-(SP)    ;* * SAVE TEST PARAMETER
11
12 172604 052737 010000 175002     BIS   #KXSNXF,@#KW.STA  ;* * SET NXM FLAG OF STATUS REGISTER
13
14 172612 022702 000007      CMP   #7,R2    ;* * IF TEST NUMBER < 7
15 172616 103404      BLO   TF100
16
17
18 172620 042737 000040 177520      BIC   #KXSDPE,@#KL.CSA
19 172626 000405      BR    TE200    ;* * THEN
20
21 172630      TE100:           ;* * ELSE
22
23 172630 052737 000040 177520      BIS   #KXSDPE,@#KL.CSA
24
25 172636 162702 000010      SUB   #10,R2    ;* * * SET ROM MAP BIT
26
27 172642      TE200:           ;* * * SUBTRACT 8 FROM TEST NUMBER
28
29 172642 006302      ASL   R2    ;* * END IF
30 172644 006302      ASL   R2    ;* * * PERFORM TEST
31 172646 004762 160000      CALL  T0(R2)
32
33 172652 042737 010000 175002      BIC   #KXSNXF,@#KW.STA  ;* * CLEAR NXM FLAG OF STATUS REGISTER
34
35 172660 012601      MOV   (SP)+,R1  ;* * RETRIEVE TEST PARAMETER
36
37 172662 012637 177524      MOV   (SP)+,3#KW.CSC
38
39 172666 011602      MOV   (SP),R2  ;* * RESTORE LED STATE
40
41 172670 010003      MOV   R0,E3  ;* * RETRIEVE TEST NUMBER (PUT LEAVE ON STACK)
42
43 172672 006302      ASL   R2  ;* * SAVE TEST RESULTS
44 172674 004772 172746      CALL  @<ERRTB-2>(R2)
45
46 172700 012602      MOV   (SP)+,R2  ;* * SET LEDS PFR RESULTS
47
48 172702 005700      TST   R0  ;* * RETRIEVE TEST NUMBER
49 172704 001410      BEQ   TE300  ;* * IF SIGNIFICANT ERROR
50
51
52 172706 005000      CLR   R0  ;* * THEN
53 172710 000261      SEC
54 172712 006100      ROL   R0
55 172714 077202      SOB   R2,TE250
56 172716 050037 175004      BIS   R0,9#KW.SC2  ;* * * SET ERROR BIT OF TEST IN DPR 2
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
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
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
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
787
788
789
789
790
791
792
793
794
795
796
797
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
817
818
819
819
820
821
822
823
824
825
826
827
827
828
829
829
830
831
832
833
834
835
836
837
837
838
839
839
840
841
842
843
844
845
846
846
847
848
848
849
849
850
851
852
853
854
855
856
857
857
858
859
859
860
861
862
863
864
865
866
866
867
868
868
869
869
870
871
872
873
874
875
876
876
877
878
878
879
879
880
881
882
883
884
885
886
886
887
888
888
889
889
890
891
892
893
894
895
895
896
896
897
897
898
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
916
917
918
918
919
919
920
921
922
923
924
925
926
926
927
928
928
929
929
930
931
932
933
934
935
936
936
937
938
938
939
939
940
941
942
943
944
945
945
946
947
947
948
948
949
949
950
951
952
953
954
955
956
956
957
958
958
959
959
960
961
962
963
964
965
965
966
967
967
968
968
969
969
970
971
972
973
974
975
975
976
977
977
978
978
979
979
980
981
982
983
984
985
985
986
986
987
987
988
988
989
989
990
991
992
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1330
1331
1331
1332
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1340
1341
1341
1342
1342
1343
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1350
1351
1351
1352
1352
1353
1353
1354
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1360
1361
1361
1362
1362
1363
1363
1364
1364
1365
1365
1366
1366
1367
1367
1368
1368
1369
1369
1370
1370
1371
1371
1372
1372
1373
1373
1374
1374
1375
1375
1376
1376
1377
1377
1378
1378
1379
1379
1380
1380
1381
1381
1382
1382
1383
1383
1384
1384
1385
1385
1386
1386
1387
1387
1388
1388
1389
1389
1390
1390
1391
1391
1392
1392
1393
1393
1394
1394
1395
1395
1396
1396
1397
1397
1398
1398
1399
1399
1400
1400
1401
1401
1402
1402
1403
1403
1404
1404
1405
1405
1406
1406
1407
1407
1408
1408
1409
1409
1410
1410
1411
1411
1412
1412
1413
1413
1414
1414
1415
1415
1416
1416
1417
1417
1418
1418
1419
1419
1420
1420
1421
1421
1422
1422
1423
1423
1424
1424
1425
1425
1426
1426
1427
1427
1428
1428
1429
1429
1430
1430
1431
1431
1432
1432
1433
1433
1434
1434
1435
1435
1436
1436
1437
1437
1438
1438
1439
1439
1440
1440
1441
1441
1442
1442
1443
1443
1444
1444
1445
1445
1446
1446
1447
1447
1448
1448
1449
1449
1450
1450
1451
1451
1452
1452
1453
1453
1454
1454
1455
1455
1456
1456
1457
1457
1458
1458
1459
1459
1460
1460
1461
1461
1462
1462
1463
1463
1464
1464
1465
1465
1466
1466
1467
1467
1468
1468
1469
1469
1470
1470
1471
1471
1472
1472
1473
1473
1474
1474
1475
1475
1476
1476
1477
1477
1478
1478
1479
1479
1480
1480
1481
1481
1482
1482
1483
1483
1484
1484
1485
1485
1486
1486
1487
1487
1488
1488
1489
1489
1490
1490
1491
1491
1492
1492
1493
1493
1494
1494
1495
1495
1496
1496
1497
1497
1498
1498
1499
1499
1500
1500
1501
1501
1502
1502
1503
1503
1504
1504
1505
1505
1506
1506
1507
1507
1508
1508
1509
1509
1510
1510
1511
1511
1512
1512
1513
1513
1514
1514
1515
1515
1516
1516
1517
1517
1518
1518
1519
1519
1520
1520
1521
1521
1522
1522
1523
1523
1524
1524
1525
1525
1526
1526
1527
1527
1528
1528
1529
1529
1530
1530
1531
1531
1532
1532
1533
1533
1534
1534
1535
1535
1536
1536
1537
1537
1538
1538
1539
1539
1540
1540
1541
1541
1542
1542
1543
1543
1544
1544
1545
1545
1546
1546
1547
1547
1548
1548
1549
1549
1550
1550
1551
1551
1552
1552
1553
1553
1554
1554
1555
1555
1556
1556
1557
1557
1558
1558
1559
1559
1560
1560
1561
1561
1562
1562
1563
1563
1564
1564
1565
1565
1566
1566
1567
1567
1568
1568
1569
1569
1570
1570
1571
1571
1572
1572
1573
1573
1574
1574
1575
1575
1576
1576
1577
1577
1578
1578
1579
1579
1580
1580
1581
1581
1582
1582
1583
1583
1584
1584
1585
1585
1586
1586
1587
1587
1588
1588
1589
1589
1590
1590
1591
1591
1592
1592
1593
1593
1594
1594
1595
1595
1596
1596
1597
1597
1598
1598
1599
1599
1600
1600

```

KXTNF - KXT11CA Native Firmware MACRO V05.00 00:51 Page 87-1
EXTEST - EXECUTE TEST

```
58 172722 010337 175006      MOV    R3,@#KW.SC3      ;* * END IF
59
60 172726                      TE300:                ;* * IF FATAL ERROR
61
62 172726 032737 000040 175002      BIT    #KXSFEF,@#KW.STA
63 172734 001401                  BEQ    TE400          ;* * THEN
64                                         ;* * STOP
65
66 172736 000777                  FTLST:: BR
67
68 172740                      TE400:                ;* * END IF
69
70 172740 042737 000040 177520      BIC    #KXSDPE,@#KW.CSA ;* * FNABLE OVERLAY 0
71
72 172746 000207                  RETURN             ;* * RETURN FROM CALL
73                                         ;* END TEST
74
75                                         ;TNERR ENTRY ADDRESS/TEST TABLE
76
77 172750 170506      ERRT3: .WORD   T1ERR
78 172752 170546      .WORD   T2ERR
79 172754 170576      .WORD   T3ERR
80 172756 170642      .WORD   T4ERR
81 172760 170659      .WORD   T5ERR
82 172762 170656      .WORD   T6ERR
83 172764 170664      .WORD   T7ERR
84 172766 170720      .WORD   T10ERR
85 172770 170726      .WORD   T11ERP
86 172772 170734      .WORD   T12ERP
87 172774 170742      .WORD   T13ERR
```

80

KXTNF - KXT11CA Native Firmware MACRO V05.00 00:51 Page 88
EXTEST - EXECUTE TEST

```
1
2
3 173000
4
5 ; Functional Description:
6
7 ; These are the entry points to the native firmware
8
9 ; Power Up entry point
10
11
12 173000 000167 000006      PERV:: JMP     PWRUP
13
14 ; Restart entry point
15
16 173004 000167 170770      RSTV:: JMP     RESTRT
17
```

116

```
1          .SBTTL PWRUP - POWER UP HANDLER
2          .ENABLE GBL,LC
3          .=173012
4          173012
5
6          ; Module name: PWRUP - POWER UP
7
8          ; System: KXT11-CA Native Firmware
9
10         ;
11         ;
12         ;
13         ;
14         ;
15         ; Functional Description:
16         ;
17         ; This routine services the Power Up Interrupt at location 173000.
18         ;
19         ; Input Parameters:
20         ;
21         ;     NONE
22         ;
23         ; Output Parameters
24         ;
25         ;     1. R5 = Top of RAM Pointer
26         ;
27         ;
28         ; Data Structures
29         ;
30         ;     1. BATTERY BACKUP POWER UP FLAG (STATUS REGISTER)
31         ;
32         ;     2. POWER UP FLAG (STATUS REGISTER)
33         ;
34         ;     3. BATTERY BACKUP FLAG 1
35         ;
36         ;     4. BATTERY BACKUP FLAG 2
37         ;
38         ;     5. Map Selection Bits (KXTCSR B, bits 1-3)
39         ;
40         ;     6. Boot Flag (BOOTFO)
41         ;
42         ; Routines Used
43         ;
44         ;     1. INIT
45         ;
46         ;     2. PTEST
47         ;
48         ;     3. BSUPV (jump to)
49         ;
50         ;     4. SETLED
51         ;
52         ;     5. TRAP24 (jump to)
```

KXTNF - KXT11CA Native Firmware MACRO V05.00 00:51 Page 90
 PWRUP - POWER UP HANDLER

120

```

1          ;* PROCEDURE POWER UP
2 173012      PWRUP:::                                ;* * IF POWER UP VECTOR ENTRY
3          ;* * THEN
4          ;* * * SET P4 = 0
5
6 173012 005004          CLR   R4
7 173014 000404          BR    PW050
8
9 173016      REINIT:::                                ;* * ELSE
10         ;* * * SET R4 = 1
11 173016 012704 000001        MOV   #1,R4
12         ;* * * SAVE DPR REGISTER 3
13 173022 013703 175006        MOV   @#KW.SC3,R3
14
15 173026      PW050:::                                ;* * END IF
16         ;* * INITIALIZE THE 8255
17 173026 012737 000212 177526        MOV   #212,0#C8255
18         ;* * CLEAR ALL KXT CSR'S
19 173034 005037 177530        CLR   @#KW.CSD
20
21 173040 012700 000020        MOV   #20,R0
22 173044 012701 175000        MOV   @#KW.CMD,R1
23 173050 005021          PW100: CLR   (R1)+t
24 173052 077002          SOB   R0,PW100
25
26 173054 005737 177562        TST   @#RBUF1A
27
28 173060 152737 000017 177524        BISB  #<KX$LD0 ! KX$LD1 ! KX$LD2 ! KX$LD3>,0#KW.CSC
29 173066 005000          CLR   R0
30 173070 077001          SOB   R0,.
31 173072 077001          SOB   R0,.
32 173074 142737 000017 177524        BICB  #<KX$LD0 ! KX$LD1 ! KX$LD2 ! KX$LD3>,0#KW.CSC
33
34 173102 012700 173112        MOV   #PW200,R0
35 173106 000167 000574          JMP   SPR5SU
36 173112      PW200:::                                ;* * IF POWER UP ENTRY
37
38 173112 005704          TST   R4
39 173114 001403          BEQ   PW300
40
41 173116 022703 000010        CMP   #8,,R3
42 173122 001013          BNE   PW500
43 173124      PW300:::                                ;* * OR REINIT ENTRY AND DPR 3 = 8
44
45
46 173124 113700 177522        MOVB  @#KL.CSB,R0
47 173130 042700 177417          BIC   #^C<KMSSWS>,R0
48 173134 006200          ASR   R0
49 173136 006200          ASR   R0
50 173140 006200          ASR   R0
51 173142 006200          ASR   R0
52 173144 010065 177774        MOV   R0,BOOTFO(R5)
53 173150 000402          BR    PW600
54
55 173152      PW500:::                                ;* * ELSE
56
57 173152 010365 177774        MOV   R3,BOOTFO(R5)

```

58 ;* * END IF
59 173156 PW600: ;* * IF POWER UP ENTRY
60
61 173156 005704 TST R4
62 173160 001034 BNE PW700 ;* * AND BATTERY BACKUP FLAG 1 = 125252
63
64 173162 022765 125252 000000 CMP #125252,BBF10(R5)
65 173170 001030 BNE PW700 ;* * AND BATTERY BACKUP FLAG 2 = 52525
66
67 173172 022765 052525 177776 CMP #52525,BBF20(R5)
68 173200 001024 BNE PW700 ;* * THEN
69 ;* * * SET BATTERY BACKUP POWER UP FLAG
70 ;* * * OF STATUS REGISTER = 1
71
72 173202 052737 002000 175002 BIS #KX\$BAT,##KW.STA
73 173210 042737 001000 175002 BIC #KX\$PWF,##KW.STA ;* * * INITIALIZE I/O
74
75 173216 004767 000060 CALL INIT ;* * * SET STATUS REGISTER TO NON NATIVE
76 ;* * * CODE STATE
77
78 173222 042737 000007 175002 BIC #KMSSTF,##KW.STA
79 173230 052737 000007 175002 BIS #KPSNNC,##KW.STA ;* * * SET/LEDS TO NON NATIVE CODE STATE DISP.
80
81 173236 012700 000000 MOV #NNCED,RO
82 173242 004767 173732 CALL SETLED ;* * * EMULATE TRAP TO 24 (NON STRUCTURED EXIT)
83
84 173246 000167 174104 JMP TRAP24
85 ;* * ELSE
86 173252 PW700: ;* * RUN SELF TEST
87
88 173252 004767 177046 CALL PTEST ;* * * INITIALIZE I/O
89
90 173256 004767 000020 CALL INIT ;* * * SET POWER UP FLAG OF STATUS REGISTER = 1
91
92 173262 052737 001000 175002 BIS #KX\$PWF,##KW.STA
93 173270 042737 002000 175002 BIC #KX\$BAT,##KW.STA ;* * * BOOT TOP (JUMP TO BOOTSTRAP SUPERVISOR)
94
95 173276 000167 175666 JMP BSUPV ;* * END IF
96 ;* END POWER UP
97

```
1          .SBTTL INIT - INITIALIZATION
2          .ENABLE LC,GBL
3
4
5          ;
6          ; Module name: INIT - INITIALIZATION
7          ;
8          ; System: KXT11-CA Native Firmware
9          ;
10         ;
11         ;
12         ;
13         ;
14         ;
15         ; Functional Description:
16         ;
17         ; This module resets and initializes the DMA controller, 7201 and
18         ; PIO. It also places the configuration data in the dual port RAM
19         ; and enables the dual port RAM.
20         ;
21         ; Input Parameters:
22         ;
23         ;      NONE
24         ;
25         ; Output Parameters
26         ;
27         ;      1. Reset DMA,7201,PIO
28         ;
29         ;      2. PIO Master Interrupt Register
30         ;
31         ;      3. " Port 1 Intr. Vec. Reg.
32         ;
33         ;      4. " " 2 " " "
34         ;
35         ;      5. " Timmer " " "
36         ;
37         ;
```

```

1                                ;* PROCEDURE INITIALIZATION
2 173302                         INIT:::                      ;* * INITIALIZE IO
3
4 173302 004767 000104          CALL   DINIT
5                                         ;* * CLEAR ALL OF STATUS REGISTER EXCEPT STATE FIELD
6                                         ;* AND POWER UP FLAGS
7 173306 042737 174770 175002      BIC    #^C<KM$STF ! KXSPWF ! KX$RAT>,@#KW.STA
8                                         ;* * IF NO AUTO SELF TEST ERRORS FOUND
9 173314 005737 175004          TST    @#KW.SC2
10 173320 001030                BNE   IN400
11                                         ;* * THEN
12                                         ;* * * PLACE TOP MAP & BOOT CONFIGURATION IN DPR REGISTER 3
13 173322 013700 177522          MOV    @#KW.CSB,RO
14 173326 042700 177401          BIC    #^C<KMSMAP ! KMSSWS>,RO
15 173332 010037 175006          MOV    RO,@#KW.SC3
16                                         ;* * * IF USER SOCKETS MAPPED TO 0
17 173336 032737 000010 177522      BIT    #KX$MP2,@#KL.CSR
18 173344 001402                BEQ   IN100
19                                         ;* * * THEN
20                                         ;* * * * TEST FOR RAM AT 0
21 173346 005000                CLR    R0
22 173350 000402                BR    IN200
23                                         ;* * * ELSE
24 173352                         IN100:
25                                         ;* * * * TEST FOR RAM AT 100000
26 173352 012700 100000          MOV    #100000,RO
27                                         ;* * * * END IF
28 173356                         IN200:
29                                         ;* * * * IF RAM PRESENT
30 173356 011001                MOV    (R0),R1
31 173360 005101                COM    R1
32 173362 005110                COM    (R0)
33 173364 020110                CMP    R1,(R0)
34 173366 001005                BNE   IN300
35                                         ;* * * THEN
36                                         ;* * * * RESTORE CONTENTS
37 173370 005101                COM    R1
38 173372 010110                MOV    R1,(R0)
39                                         ;* * * * SET RAM PRESENT BIT OF DPR REG. 3
40 173374 052737 000001 175006      BIS    #BIT0,@#KW.SC3
41                                         ;* * * END IF
42 173402                         IN300:
43                                         ;* * END IF
44 173402                         IN400:
45                                         ;* * ENABLE THE DUAL PORT RAM
46 173402 052737 000100 177530      BTS    #KX$DEN,@#KW.CSD
47                                         ;* * RETURN
48 173410 000207                RETURN
49                                         ;* END INITIALIZATION

```

```
1 .SBTTL DINIT - DEVICE INITIALIZATION
2 .ENABLE LC, GBL
3
4
5 ; Module name: DINIT - I/O DEVICE INITIALIZATION
6 ; System: KXT11-CA Native Firmware
7 ;
8 ;
9 ;
10 ;
11 ;
12 ;
13 ; Functional Description:
14 ;
15 ; Initialize all programmable I/O devices. Disables interrupts and sets
16 ; programmable interrupt vector addresses.
17 ;
```

124

```
1 ;* PROCEDURE DINIT
2 173412 DINIT:: ;* * RESET DMA
3
4 173412 005037 174454 CLR @#D4ACDA ;* * RESET 7201
5
6 173416 113700 175700 MOVB @#SL2SAA,R0
7 173422 112737 000030 175704 MOVR #30,@#SL2CAA
8 173430 112737 000030 175714 MOVB #30,@#SL2CBA
9 173436 112737 000002 175704 MOVB #2,@#SL2CAA
10 173444 112737 000024 175704 MOVR #24,@#SL2CAA
11 173452 112737 000001 175714 MOVB #1,@#SL2CBA ;(SET STATUS EFFECTS VECTOR BIT)
12 173460 112737 000004 175714 MOVB #4,@#SL2CBA
13
14 173466 112737 000001 177000 MOVB #01,@#PIOICA
15 173474 105037 177000 CLRB @#PIOICA ;* * RESET PIO
16
17 173500 112737 000200 177004 MOVB #200,@#PIOAVA ;* * SETUP PIO PORT A INTERRUPT VECTOR REGISTER
18
19 173506 112737 000204 177006 MOVB #204,@#PIOBVA ;* * SETUP PIO PORT B INTERRUPT VECTOR REGISTER
20
21 173514 112737 000210 177010 MOVB #210,@#PIOTVA ;* * SETUP PIO TIMER INTERRUPT VECTOR REGISTER
22
23 173522 112737 000110 174532 MOVB #110,@#DMAOVA ;* * SETUP DMA CHANNEL 0 INTERRUPT VECTOR REGISTER
24
25 173530 112737 000114 174530 MOVB #114,@#DMAIVA ;* * SETUP DMA CHANNEL 1 INTERRUPT VECTOR REGISTER
26
27 173536 000207 RETURN ;* END DINIT
```

KXTNF - KXT11CA Native Firmware MACRO V05.00 00:51 Page 95
ACCUM - ACCUMULATE NUMBER

```
1          .SBTTL ACCUM - ACCUMULATE NUMBER
2          .ENABLE LC, GBL
3
4
5          ; Module name: ACCUMULATE - ACCUM
6
7          ; System: KXT11-CA Native Firmware
8
9
10         ;
11         ;
12         ;
13         ; Functional Description:
14
15         ; This routine accumulates a 16 bit number by shifting the previous
16         ; accumulation left 3 and "or"ing the new input value (which is a single
17         ; octal digit value) into the accumulation.
18
19
20         ; Input Parameters:
21
22         ;      1) R0 = OCTAL DIGIT INPUT
23         ;      2) R4 = PREVIOUS ACCUMULATED NUMBER
24
25         ; Output Parameters:
26
27         ;      1) R4 = ACCUMULATED NUMBER
28
29         ; Routines Used:
30
31         ;      None
```

125

KXTNF - KXT11CA Native Firmware MACRO V05.00 00:51 Page 96
ACCUM - ACCUMULATE NUMBER

```

1                               ;* PROCEDURE ACCUMULATE
2 173540          ACCUM::           ;* * SHIFT INPUT LEFT 3
3
4 173540  006304      ASL    R4
5 173542  006304      ASL    R4
6 173544  006304      ASL    R4
7
8 173546  050004      BIS    R0,R4 ;* * SET ACCUMULATED NUMBER = ACCUMULATED NUMBER (OR) DATA
9
10 173550 000207     RETURN        ;* END ACCUMULATE

```

KXTNF - KXT11CA Native Firmware MACRO V05.00 00:51 Page 97
GETIN - GET INPUT

```
1 .SBTTL GFTIN - GET INPUT
2 .ENABLE LC,GBL
3
4 ; Module name: GET INPUT - GETIN
5 ; System: KXT11-CA Native Firmware
6 ;
7 ;
8 ;
9 ;
10 ;
11 ;
12 ;
13 ; Functional Description:
14 ;
15 ; This routine inputs a character and sets the non numeric flag if the
16 ; input is not a number. If the input is a number it is converted to binary,
17 ; clears the non-numeric flag and sets the data received flag.
18 ;
19 ; Output Parameters:
20 ;
21 ; 1) Data Received flag - DARFCB of R3
22 ; 2) Non Numeric Flag - Carry Flag
23 ; 3) Input Data, - R0
```

KXTNF - KXT11CA Native Firmware MACRO V05.00 00:51 Page 98
GETIN - GET INPUT

```
1
2 173552
3
4
5 173552 004767 000036      GETIN::                      ;* PROCEDURE GET INPUT
6
7 173556 103375              GE100: CALL    CHKCHR          ;* * DO UNTIL.INPUT
8
9 173560 122700 000060      BCC     GE100            ;* * END UNTIL
10 173564 101011             CMPB    #0,R0           ;* * IF INPUT = 0 THRU 7
11 173566 122700 000067      BHI     GE200
12 173572 103406             CMPB    #7,R0
13
14
15 173574 162700 000060      BLO     GE200            ;* * THEN
16
17 173600 052703 000002      SUB    #60,R0           ;* * * CONVERT TO BINARY
18
19 173604 000241             BIS    #DARFCB,R3        ;* * * SET DATA RECEIVED FLAG
20 173606 000401             CLC
21
22 173610
23
24 173610 000261             BR     GE300            ;* * * CLEAR NON NUMERIC FLAG
25
26 173612
27
28 173612 000207             GE200: SEC
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
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
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
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
687
688
689
689
690
691
692
693
694
695
696
697
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
717
718
719
719
720
721
722
723
724
725
726
727
727
728
729
729
730
731
732
733
734
735
736
737
737
738
739
739
740
741
742
743
744
745
746
746
747
748
748
749
749
750
751
752
753
754
755
756
757
757
758
759
759
760
761
762
763
764
765
766
766
767
768
768
769
769
770
771
772
773
774
775
776
776
777
778
778
779
779
780
781
782
783
784
785
786
786
787
788
788
789
789
790
791
792
793
794
795
796
796
797
798
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
817
818
819
819
820
821
822
823
824
825
826
827
827
828
829
829
830
831
832
833
834
835
836
837
837
838
839
839
840
841
842
843
844
845
846
846
847
848
848
849
849
850
851
852
853
854
855
856
857
857
858
859
859
860
861
862
863
864
865
866
866
867
868
868
869
869
870
871
872
873
874
875
876
876
877
878
878
879
879
880
881
882
883
884
885
886
886
887
888
888
889
889
890
891
892
893
894
895
895
896
897
897
898
898
899
899
900
901
902
903
904
905
906
907
907
908
909
909
910
911
912
913
914
915
915
916
917
917
918
918
919
919
920
921
922
923
924
925
926
926
927
928
928
929
929
930
931
932
933
934
935
936
936
937
938
938
939
939
940
941
942
943
944
945
945
946
947
947
948
948
949
949
950
951
952
953
954
955
955
956
957
957
958
958
959
959
960
961
962
963
964
965
965
966
967
967
968
968
969
969
970
971
972
973
974
975
975
976
977
977
978
978
979
979
980
981
982
983
984
985
985
986
987
987
988
988
989
989
990
991
992
993
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1330
1331
1331
1332
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1340
1341
1341
1342
1342
1343
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1350
1351
1351
1352
1352
1353
1353
1354
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1360
1361
1361
1362
1362
1363
1363
1364
1364
1365
1365
1366
1366
1367
1367
1368
1368
1369
1369
1370
1370
1371
1371
1372
1372
1373
1373
1374
1374
1375
1375
1376
1376
1377
1377
1378
1378
1379
1379
1380
1380
1381
1381
1382
1382
1383
1383
1384
1384
1385
1385
1386
1386
1387
1387
1388
1388
1389
1389
1390
1390
1391
1391
1392
1392
1393
1393
1394
1394
1395
1395
1396
1396
1397
1397
1398
1398
1399
1399
1400
1400
1401
1401
1402
1402
1403
1403
1404
1404
1405
1405
1406
1406
1407
1407
1408
1408
1409
1409
1410
1410
1411
1411
1412
1412
1413
1413
1414
1414
1415
1415
1416
1416
1417
1417
1418
1418
1419
1419
1420
1420
1421
1421
1422
1422
1423
1423
1424
1424
1425
1425
1426
1426
1427
1427
1428
1428
1429
1429
1430
1430
1431
1431
1432
1432
1433
1433
1434
1434
1435
1435
1436
1436
1437
1437
1438
1438
1439
1439
1440
1440
1441
1441
1442
1442
1443
1443
1444
1444
1445
1445
1446
1446
1447
1447
1448
1448
1449
1449
1450
1450
1451
1451
1452
1452
1453
1453
1454
1454
1455
1455
1456
1456
1457
1457
1458
1458
1459
1459
1460
1460
1461
1461
1462
1462
1463
1463
1464
1464
1465
1465
1466
1466
1467
1467
1468
1468
1469
1469
1470
1470
1471
1471
1472
1472
1473
1473
1474
1474
1475
1475
1476
1476
1477
1477
1478
1478
1479
1479
1480
1480
1481
1481
1482
1482
1483
1483
1484
1484
1485
1485
1486
1486
1487
1487
1488
1488
1489
1489
1490
1490
1491
1491
1492
1492
1493
1493
1494
1494
1495
1495
1496
1496
1497
1497
1498
1498
1499
1499
1500
1500
1501
1501
1502
1502
1503
1503
1504
1504
1505
1505
1506
1506
1507
1507
1508
1508
1509
1509
1510
1510
1511
1511
1512
1512
1513
1513
1514
1514
1515
1515
1516
1516
1517
1517
1518
1518
1519
1519
1520
1520
1521
1521
1522
1522
1523
1523
1524
1524
1525
1525
1526
1526
1527
1527
1528
1528
1529
1529
1530
1530
1531
1531
1532
1532
1533
1533
1534
1534
1535
1535
1536
1536
1537
1537
1538
1538
1539
1539
1540
1540
1541
1541
1542
1542
1543
1543
1544
1544
1545
1545
1546
1546
1547
1547
1548
1548
1549
1549
1550
1550
1551
1551
1552
1552
1553
1553
1554
1554
1555
1555
1556
1556
1557
1557
1558
1558
1559
1559
1560
1560
1561
1561
1562
1562
1563
1563
1564
1564
1565
1565
1566
1566
1567
1567
1568
1568
1569
1569
1570
1570
1571
1571
1572
1572
1573
1573
1574
1574
1575
1575
1576
1576
1577
1577
1578
1578
1579
1579
1580
1580
1581
1581
1582
1582
1583
1583
1584
1584
1585
1585
1586
1586
1587
1587
1588
1588
1589
1589
1590
1590
1591
1591
1592
1592
1593
1593
1594
1594
1595
1595
1596
1596
1597
1597
1598
1598
1599
1599
1600
1600
1601
1601
1602
1602
1603
1603
1604
1604
1605
1605
1606
1606
1607
1607
1608
1608
1609
1609
1610
1610
1611
1611
1612
1612
1613
1613
1614
1614
1615
1615
1616
1616
1617
1617
1618
1618
1619
1619
1620
1620
1621
1621
1622
1622
1623
1623
1624
1624
16
```

127

```
1 .SBTTL CHKCHR - CHECK CHARACTER
2 .ENABLE GBL,LC
3
4 ; Module name: CHKCHR - CHECK CHARACTER
5 ; System: Micro-PAX RTS
6 ;
7 ;
8 ;
9 ;
10 ;
11 ;
12 ;
13 ;
14 ; Functional Description:
15 ;
16 ; This module tests the console serial line receive buffer for a
17 ; character, signals if one was received, echos the character
18 ; and returns that character to the calling routine.
19 ; This module is to be used for input polling. It
20 ; will handle XON XOFF protocol. This routine will not echo
21 ; the XON or XOFF characters.
22
23 ;INPUT PARAMETERS
24
25 ;      NONE
26
27 ;OUTPUT PARAMETERS
28
29 ;      1) CARRY BIT = 1 FOR CHARACTER RECEIVED, = 0 FOR NO CHARACTER
30 ;          2) R0 = RECEIVE BUFFER DATA, IF DATA RECEIVED
31
32 ;DATA STRUCTURES USED
33
34 ;      1) SLU1 CSR
35 ;          2) SLU1 RECEIVE BUFFER
36
37 ;ROUTINES USED
38
39 ;      NONE
```

1	000021	XON	=	*Q-100
2	000023	XOFF	=	*S-100

KXTMF - KXT11CA Native Firmware MACRO V05.00 00:51 Page 101
 CHKCHR - CHECK CHARACTER

128

```

1          ;* PROCEDURE CHECK CHARACTER
2 173614      CHKCHR:::             CLC
3 173614 000241
4
5 173616 105737 177560           TSTR  @#RCSR1A
6 173622 100030                 SPL   CH400
7
8
9 173624 113700 177562           MOVB  @#RBUF1A,R0
10
11 173630 042700 000200          BTC   #BIT7,R0
12
13
14 173634 122700 000023          CMPB  #XOFF,R0
15 173640 001004                 BNE   CH100
16
17 173642 052703 000040          BIS   #XOFFGB,R3
18 173646 000241                 CLC
19 173650 000415                 BR    CH300
20
21 173652      CH100:             CMPB  #XON,R0
22 173652 122700 000021          BNE   CH200
23 173656 001004
24
25 173660 042703 000040          BIC   #XOFFGB,R3
26 173664 000241                 CLC
27 173666 000406                 BR    CH300
28
29 173670      CH200:             CMPB  #LF,R0
30
31 173670 122700 000012          BEQ   CP250
32 173674 001402
33
34
35 173676 010037 177566          MOV   R0,@#XBUF1A
36
37 173702      CH250:             SEC
38
39 173702 000261
40
41 173704      CH300:             RETRN
42
43 173704 000207
44 173704

```

;

** * IF RECEIVE DONE

** * THEN

** * * READ RECEIVE BUFFER

** * * CLEAR PARITY BIT

** * * CASE CHARACTER

** * * WHEN XOFF

** * * * SET XOFF FLAG = 1

** * * WHEN XON

** * * * SET XOFF FLAG = 0

** * * ELSE

** * * * IF CHARACTER NOT LF

** * * * THEN

** * * * * ECHO CHARACTER

** * * * END IF

** * * * SIGNAL CHARACTER RECEIVED

** * * END CASE

** * END IF

```
1           .SBTTL SPR5SU - SP/R5 SETUP
2           .ENABLE LC,CBL
3
4           ;
5           ; Module name: SP AND R5 SETUP - SPR5SU
6           ;
7           ; System: KXT11-CA Native Firmware
8           ;
9           ;
10          ;
11          ;
12          ;
13          ;
14          ; Functional Description:
15          ;
16          ; This module sets up the SP to the native firmware stack area and
17          ; setting R5 to the address of the last word of native RAM.
18          ;
19          ;
20          ; Input Parameters:
21          ;
22          ;   1) Return address in R0
23          ;
24          ; Output Parameters:
25          ;
26          ;   1) Top of RAM address in R5
27          ;   2) Stack Pointer
28
```

1
2
3 173706 SPR5SU: ;* PROCEDURE SP AND R5 SETUP
4 ;* * CASE MAP SELECTION
5 ;* * WHEN MAP BIT 2 = 0
6 173706 132737 000010 177522 BITB #KXSMR2,0#KW.CSB
7 173714 001003 BNE SP200
8 ;* * * SET STACK POINTER TO STACK 1
9 173716 012706 077762 MOV #SP1,SP
10 173722 000415 BR SP400
11 ;* * WHEN MAP BITS 0 AND 1 = 11
12 173724 132737 000002 177522 SP200: BITB #KXSMR0,0#KW.CSB
13 173732 001407 BEQ SP300
14 173734 132737 000004 177522 BITB #KXSMR1,0#KW.CSB
15 173742 001403 BEQ SP300
16 ;* * * SET STACK POINTER TO STACK 3
17 173744 012706 157762 MOV #SP3,SP
18 173750 000402 BR SP400
19 ;* * ELSE
20 173752 SP300: ;* * * SET STACK POINTER TO STACK 2
21 22 173752 012706 137762 MOV #SP2,SP
23 ;* * END CASE
24 173756 SP400: ;* * SET R5 = TOP OF RAM (SP + STACK OFFSET)
25 26 173756 010605 MOV SP,R5
27 173760 062705 000014 ADD #SPOFF,R5
28 ;* * RETURN TO CALLER
29 173764 000110 JMP (R0)
30 ;* END SP AND R5 SETUP

KXTNF - KXT11CA Native Firmware MACRO V05.00 00:51 Page 104
VERS - VERSION NUMBER AND TEST

```
1 .SBTTL VERS - VERSION NUMBER AND TEST
2     173774      .=173774
3             .ENABLE LC
4
5 ; Module name: VERS - VERSION NUMBER
6 ; System: KXT11-CA Native Firmware
7 ;
8 ;
9 ;
10 ;
11 ;
12 ;
13 ;
14 ;
15 ; Functional Description:
16 ;
17 ;This module places the version number in the last word of the
18 ;permanently mapped native ROM.
19
20 173774    126    061    056 VERS:: .ASCII "V1.0"
```

KXTNF - KXT11CA Native Firmware MACRO V05.00 00:51 Page 105
SELF TEST OVERLAY 1

```
I 131
1 .SBTTL SELF TEST OVERLAY 1
2           .LIST ME,MEB,SEQ,LOC,BIN      ; NORMAL LISTING MODE
3           .NLIST MC,CND,BEX          ; DITTO
4           .ENABL AMA,GBL
5     174000      .=174000
6
7 ; Module name: SELF TEST OVERLAY 1
8 ; System: KXT11-CA Native Firmware
9 ;
10 ;
11 ;
12 ;
13 ;
14 ;
15 ;
16 ;
17 ; Functional Description:
18 ;
19 ; This module contains overlay 1 of the self test code. The self test
20 ; modules in this overlay are:
21 ;
22 ;     Parallel I/O port Z8036,
23 ;     DMA controller Z8016,
24 ;     Q-BUS interrupts, and
25 ;     Two-port RAM.
26 ;
27 ; Overlay 1 physically occupies the top 2 KB of the boot ROM. It
28 ; executes starting at 160000 when the DIAG PROM FN bit, bit 5, in
29 ; KXTCSRA is 1.
```

1 ; SCRATCH RAM ALLOCATION.
2 ;
3 ; R5 IS ODT'S "TOP-OF-MEM" POINTER, AND IS PRESERVED AS SUCH.
4 ; ALL SCRATCH REFERENCES ARE INDEXED FROM R5 AS FOLLOWS:
5 ;
6 ;
7 ;
8 000005 STOP= 35 ; 077776 137776 157776
9 ;
10 150002 BUFR1= -27776 ; 050000 110000 130000 TOP-6KW
11 160002 BUFR2= -17776 ; 060000 120000 140000 TOP-4KW
12 170002 TMP1= -07776 ; 070000 130000 150000 TOP-2KW
13 170004 TMP2= TMP1+2 ;
14 170006 TMP3= TMP1+4 ;
15 170010 TMP4= TMP1+6 ;
16 170012 TMP5= TMP1+10 ; > SCRATCH REGISTERS...
17 170014 TMP6= TMP1+12 ; / ...AND STACK SPACE.
18 170016 TMP7= TMP1+14 ;
19 170020 TMP8= TMP1+16 ;
20 177776 \$EF1= -2 ; LAST 2 WORDS OF NATIVE RAM (BATTERY-BACK-FLAGS)...
21 000005 \$EF= STOP ;...ARE UTILIZED AS OUR ERROR FLAGS.
22 ;
23 ; MISCELLANEOUS STUFF.
24 ;
25 000007 STN= 7 ; INIT TEST NUMBER SEQUENCE.
26 014000 K3K= 3072.*2 ; HISEG CALLS REQUIRE 3KW OFFSET.
27 ;
28 ; A FEW MACROS.
29 ;
57 ;
78 ;
91 ;
97 ;
98 ;
99 .NLIST MD

```
1 ; A FEW DEFINITIONS
2 ;
3 ; STANDARD STUFF.
4 ;
5      000000      PRO= 000          ; THE PRIORITIES.
6      000040      PR1= 040
7      000100      PR2= 100
8      000140      PR3= 140
9      000200      PR4= 200
10     000240      PR5= 240
11     000300      PR6= 300
12     000340      PR7= 340
13
14 ; PRIMARY CONTROL REGISTERS.
15 ;
16     177560      $SL1= 177560      ; DEC DC319 (DLART) ASYNC SERIAL PORT.
17     175700      $SL2= 175700      ; NEC7201 SYNC/ASYNC SERIAL PORT.
18     175720      $I8254= 175720    ; I8254 PROGRAMMABLE BAUD GEN FOR ABOVE.
19     177000      $PIO= 177000      ; ZR036 PIO/PIT.
20     174400      $DMA= 174400      ; AM28016 DMA CONTROLLER.
21     177520      $CSRA= 177520      ;
22     177522      $CSRB= 177522      ; > SECONDARY CSR'S...
23     177524      $CSPC= 177524      ; / ...EMBEDDED IN I8255,CHIP.
24     177526      $CSRCON= 177526    ;
25     177530      $CSR= 177530      ; PRIMARY CSR.
26     177532      $QIR= 177532      ; Q-BUS INTERRUPT REGISTER.
27     175000      $DPR= 175000      ; DUAL-PORT RAM (LOCAL).
28     160000      $DOPR1= 160000    ; DUAL-PORT RAM (GLOBAL BASE ID 0-7...
29     175400      $DOPR2= 175400    ; ...AND ID 8-F).
30
31     175000      $IPV= $DPR        ; IPV INTERRUPTS (VIA DPR WORDS 0, 4, 8, AND 12).
32     177524      $LEDS= $CSRC       ; CSRC<3:0> DRIVE THE LEDS.
33     177524      $UID= $CSRC       ; CSRC<7:4> SHOW THE ID SWITCH SETTING.
```

KXTNF - KXT11CA Native Firmware MACRO V05.00 00:51 Page 108
SELF TEST OVERLAY 1

```

1 ; DEFAULT VECTORS.
2 ;
3     000024 PWRV= 024 ; PWR-UP (BRESET) (NON-MASKABLE).
4     000060 SL1RV= 060 ; CONSOLE (SLU1) RCVR (PRI 4).
5     000064 SL1XV= 064 ; CONSOLE (SLU1) XMTR (PRI 4).
6     000070 NECV= 070 ; SYNC/ASYNC BOTH CHANS (PRI 4).
7     000100 BEVNT= 100 ; LINE CLOCK (PRI 6).
8     000104 PEVNT= 104 ; PROGRAMMABLE CLOCK(S) (PRI 6).
9         ; OPEN
10    000114 PRTYV= 114 ; MEMORY PARITY (PRI 7).
11    000120 DPRV4= 120 ; DUAL-PORT RAM WORD 4 (PRI 5).
12    000124 DPRV8= 124 ; DUAL-PORT RAM WORD 8 (PRI 5).
13    000130 BIACKV= 130 ; Q-BUS "IACK" (PRI 5).
14    000134 DPRV12= 134 ; DUAL-PORT-RAM WORD 12 (PRI 5).
15         ; 140 ; RESERVED (FALCON).
16    000144 QIRV= 144 ; Q-BUS REQUEST (ARBITER EXECUTES RTI).
17    000150 QIRV1= 150 ; DITTO (ARBITER EXECUTES RESET,RTI)
18    000154 PIOAV= 154 ; PIO PORT A (PRI 4).
19    000160 PIOBV= 160 ; PIO PORT B (PRI 4).
20    000164 DMAV= 164 ; DMA (BOTH CHANNELS) (PRI 4).

```

134

KXTNF - KXT11CA Native Firmware MACRO V05.00 00:51 Page 109
SELF TEST OVERLAY 1

```

1 ; HIGH SEGMENT ENTRY BLOCK
2 ;
3 ; CALL: BIS #BITS,CSRA ; ENABLE HIGH SEGMENT.
4 ; CALL TN ; CALL TEST 10-17.
5 ; BEQ OK ; OR BNE ERROR.
13    160000 T10==.-K3K
        174000 000137 160040 JMP    TST10-K3K    ;; EXECUTE TEST 10.
        160004 T11==.-K3K
        174004 000137 161106 JMP    TST11-K3K    ;; EXECUTE TEST 11.
        160010 T12==.-K3K
        174010 000137 162102 JMP    TST12-K3K    ;; EXECUTE TEST 12.
        160014 T13==.-K3K
        174014 000137 162432 JMP    TST13-K3K    ;; EXECUTE TEST 13.
        160020 T14==.-K3K
        174020 005000 CLR    R0      ;; SINCE THERE IS NO TEST BY THIS NAME, SHOW NO ERRORS.
        174022 000207 RRETURN
        160024 T15==.-K3K
        174024 005000 CLR    R0      ;; SINCE THERE IS NO TEST BY THIS NAME, SHOW NO ERRORS.
        174026 000207 RETURN
        160030 T16==.-K3K
        174030 005000 CLR    R0      ;; SINCE THERE IS NO TEST BY THIS NAME, SHOW NO ERRORS.
        174032 000207 RETURN
        160034 T17==.-K3K
        174034 005000 CLR    R0      ;; SINCE THERE IS NO TEST BY THIS NAME, SHOW NO ERRORS.
        174036 000207 RETURN

```

```

1          .S8TTL - T10  PARALLEL I/O PORT Z8036
2          ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
3          ;;; TEST 10 -- PARALLEL I/O PORT Z8036
4          ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
5          174040          TST10:
6          174040          TPIO:
7          174040          013746 000104      MOV     PEVNT,-(SP)      ;; SAVE PEVNT.
8          174044          013746 000106      MOV     PEVNT+2,-(SP)
9          174050          013746 000154      MOV     PIOAV,-(SP)      ;; SAVE PIOAV.
10         174054          013746 000156      MOV     PIOAV+2,-(SP)
11         174060          013746 000160      MOV     PIORV,-(SP)      ;; SAVE PIORV.
12         174064          013746 000162      MOV     PIORV+2,-(SP)
13         174070          005015          CLR     (SEF)           ;; CLEAR ERROR FLAG.
14
15
16
17
18
19 174072 004737 163602          CALL    XRAMLO-K3K      ; VECTOR SPACE USABLE ???
20 174076 001404          BFQ    4$              ; PROCEED IF SO.
21 174100 052715 000001          BIS    #BIT0,(SEF)      ;; EO = ROM IN VECTOR SPACE -- CAN'T RUN
22 174104 000137 160746          JMP    PTOXIT-K3K
23
24 174110 012703 000060          4$:   MOV    #48,,R3        ; CHECK 48 REGISTERS...
25 174114 012704 177000          MOV    #SPIO,R4        ;...STARTING HERE.
26 174120 112714 000001          MOVB   #1,(R4)        ; SET "CHIP-RESET" IN MICR.
27 174124 122714 000001          CMPPR #1,(R4)        ; READ IT, SHOULD SEF THE 1 BIT.
28 174130 000403          BR    2$              ; CHECK IT AT 2$.
29
30 174132 112714 177777          1$:   MOVR  #-1,(R4)      ; THEN, WRITES SHOULD BE IGNORED...
31 174136 105714          TSTR   (R4)          ;...AND READS RETURN ZERO.
32 174140 001402          2$:   BEQ    3$              ; BR IF OK.
33 174142 052715 000002          BIS    #BIT1,(SEF)      ;; E1 = RESET STATE INCORRECT
34 174146 005724          3$:   TST    (R4)+        ; NEXT ADDRESS...
35 174150 077310          SOB    R3,1$          ;...AND LOOP 'TIL DONE.
36
37
38
39 000000          MIC=   000          ; MASTER INTERRUPT CONTROL REG.
40 000002          MCC=   002          ; MASTER CONFIGURATION CONTROL REG.
41 000010          CTV=   010          ; TIMER(S) INTERRUPT VECTOR REG.
42 000070          CTMS=  070          ; 1ST (OF 3) TIMER MODE SPEC REG.
43 000024          CTCS=  024          ; 1ST (OF 3) TIMER CMD/STATUS REG.
44 000054          CTTC=  054          ; 1ST (OF 6) TIME CONSTANT REG (3 PAIR).
45 000040          CTCC=  040          ; 1ST (OF 6) CURRENT COUNT REG (3 PAIR).

```

```

46 ; PIT: MOV #$PIO,R4 ; GET BASE ADDRESS.
47 174152 012704 177000 BICR #1,(R4) ; CLEAR "CHIP-RESET"...
48 174156 142714 000001 MOV #48,,R3
49 174162 012703 000060 CLR (R4)+ ;...AND CLEAR ALL REGISTERS.
50 174166 005024 SOB R3,2$ ; GET BASE AGAIN.
51 174170 077302 MOV #$PIO,R4 ; SET "MASTER INT ENAB".
52 174172 012704 177000 MOVB #200,(R4) ; SET TO VECTOR THRU "PEVNT".
53 174176 112714 000200 MOVB #PEVNT,CTV(R4) ; ENABLE ALL 3 TIMERS...
54 174202 112764 000104 000010 MOVE #160,MCC(R4) ;...CTMS = 0 = PULSE-OUT, ONCE-ONLY...
55 174210 112764 000160 000002 ;...CTTC = 0 = TIME K 65536.
56 ; NOW RAISE R4 TO 1ST TIMER CSR.
57
58 174216 062704 000024 ADD #CTCS,R4 ; RAISE CPU TO PIO LEVEL.
59
60 174222 106427 000200 PIT1: MTPS #PR4 ; SET VECTOR.
61 174226 012737 160320 000104 MOV #3S-K3K,PEVNT
62 174234 012737 000300 000106 MOV #PR6,PEVNT+2
63 174242 005000 CLR R0 ; SET A 1 SECOND KEEP-ALIVE TIMER.
64 174244 012701 000004 MOV #4,R1 ; CLEAR "IUSIIP"...
65 174250 112714 000040 MOVB #40,,(R4) ;...AND SET "IEGCBITC8"...
66 174254 112714 000306 MOVR #306,(R4) ;...TIMER SHOULD TAKE-OFF !!
67 174260 000240 NOP ; TIMER RUNNING (CIP=1) ??
68 174262 132714 000001 1$: BITB #1,(R4), ; BR IF SO.
69 174266 001004 BNE 2$ ; ABORT.
70 174270 077004 SOB R0,1$ ; E2 = TIMER DIDN'T START
71 174272 052715 000004 BTS #BIT2,($EF)
72 174276 000426 BR 7$ ; ABORT.
73 174300 132714 000001 2$: BITB #1,(R4) ; TIMER DONE (CIP=0) ??
74 174304 001411 BEQ 4$ ; BR IF SO.
75 174306 077004 SOB R0,2$ ; E3 = TIMER NEVER STOPS.
76 174310 077105 SOB R1,2$ ; ABORT.
77 174312 052715 000010 BIS #BIT3,($EF)
78 174316 000416 BR 7$ ; INTERRUPT -- SHOULD HAVE BEEN MASKED.
79
80 174320 000240 3$: NOP ; E4 = INTERRUPT NOT MASKED AT LEVEL 4
81 174322 052715 000020 BIS #BIT4,($EF)
82 174326 000411 BR 6$ ; NOW CHANGE THE VECTOR...
83
84 174330 062737 000032 000104 4$-5$: ADD #6S-3$,PEVNT ; LOWER THE CPU...
85 174336 106427 000000 MTPS #PK0 ;...AND LET INTERRUPT COME IN.
86 174342 000240 NOP ; E5 = INTERRUPT NOT RECEIVED.
87 174344 052715 000040 BIS #BITS,($EF)
88 174350 000401 BR 7$ ; INTERRUPT -- FIX THE STACK.
89
90 174352 022626 6$: CMP (SP)+,(SP)+ ; CLEAR "IUSIIP!GCB"...
91 174354 012714 000040 MOV #40,(R4) ;...AND "IE" (TIMER OFF).
92 174360 012714 000160 MOV #160,(R4) ; BUMP TO NEXT TIMER CHD/STAT...
93 174364 005724 TST (R4)+ ;...AND LOOP 'TIL ALL 3 DONE.
94 174366 020427 177032 CMP R4,#$PIO+CTCS+6
95 174372 103713 BLO PTT1 ; THEN, RESET THE CHIP.
96 ; FALL THRU TO DATA TEST.
97 174374 112737 000001 177000 MOVB #1,$PIO
98 174402 000240 NOP

```

137

```

1 ; DATA PORT(S) TEST.
2 ; *** REQUIRES EXTERNAL LOOP-BACK ***
3 ;
4
5
6      000004      PAV=    004      ; PORT A VECTOR...
7      000020      PACS=   020      ;...CMD/STATUS...
8      000032      PADB=   032      ;...DATA BUFFER...
9      000100      PAMS=   100      ;...MODE...
10     000102      PAHS=   102      ;...AND HANDSHAKE SPEC.
11     000006      PRV=    006      ; PORT B VECTOR...
12     000022      PRCS=   022      ;...CMD/STATUS...
13     000034      PBDB=   034      ;...DATA BUFFER...
14     000120      PBMS=   120      ;...MODE...
15     000122      PBHS=   122      ;...AND HANDSHAKE SPEC.
16     000014      PCDD=   014      ; PORT C DATA DIRECTION...
17     000036      PCDE=   036      ;...AND BUFFER (HANDSHAKE BITS).
18     000140      XCVR=   140      ; 1ST WORD ABOVE THE CHIP HOLDS...
19     164377      AI.B0= <350*400>1377 ;...XCVR CONTROL BITS TTL/A(IN)IB(OUT)-
20     165400      AO.BI= <353*400>10 ;       DITTO          TTL/A(OUT)IB(IN).
21
22     170002      BYTES= TMP1      ; DEFINE A SCRATCH LOCATION.
23
24 174404 012704 177000  PIOD1: MOV #SP10,R4      ; GET BASE ADDRESS.
25 174410 142714 000001  BTCR #1,(R4)      ; CLEAR "CHIP-RESET".
26 174414 012701 000154  MOV #PIOAV,R1
27 174420 010164 000004  MOV R1,PAV(R4)      ; SET PORT A VECTORS.
28 174424 012721 161010  MOV #AI-K3K,(R1)+ 
29 174430 012721 000200  MOV #PK4,(R1)+ 
30 174434 010164 000006  MOV R1,PBV(R4)      ; SET PORT B VECTORS.
31 174440 012721 161024  MOV #BI-K3K,(R1)+ 
32 174444 012721 000200  MOV #PR4,(R1)+ 
33 174450 112764 000100 000100  PIOD1: MOVB #100,PAMS(R4)      ; SET PORT A INPUT...
34 174456 112764 000000 000102  MOVB #000,PAHS(R4)      ;...INTERLOCKED HS...
35 174464 112764 000300 000020  MOVB #300,PACS(R4)      ;...INT ENABLED.
36 174472 112764 000200 000120  MOVB #200,PBMS(R4)      ; SET PORT B OUTPUT...
37 174500 112764 000000 000122  MOVB #000,PBHS(R4)      ;...INFRLOCKED HS...
38 174506 112764 000300 000022  MOVB #300,PBCS(R4)      ;...INT ENABLED.
39 174514 112764 000200 000022  MOVB #200,PBGS(R4)      ; SET OUT "IP" TO PRIME THE LOOP.
40 174522 012764 164377 000140  MOV #AI.B0,XCVR(R4)      ; SET XCVR CONTROL BITS.
41 174530 000414      BR PIOD3
42 174532 112764 000200 000100  PIOD2: MOVB #200,PAMS(R4)      ; CHANGE POPT A TO OUTPUT...
43 174540 112764 000200 000020  MOVB #200,PACS(R4)      ;...AND SET OUT "IP".
44 174546 112764 000100 000120  MOVB #100,PBMS(R4)      ; CHANGE PORT B TO INPUT.
45 174554 012764 165400 000140  MOV #AO.B1,XCVR(R4)      ; CHANGE XCVR CONTROLS.
46 174562 106427 000200      PIOD3: MTPS #PR4      ; RAISE CPU.
47 174566 112764 000377 000014  MOVB #377,PCDD(R4)      ; PORT C DIRECTION ALWAYS "IN".
48 174574 012702 163646  MOV #XFLT10-K3K,R2      ; XMT BUFFER POINTER => R2.
49 174600 010503      MOV STOP,R3
50 174602 062703 150002  ADD #BUFR1,R3      ; RCV BUFFER POINTER => R3.
51 174606 010346      MOV R3,-(SP)      ; SAVE A COPY OF R3.
52 174610 012700 000020  MOV #16.,R0
53 174614 005023      IS: CLR (R3)+      ; CLEAR IT...
54 174616 077002      SOB R0,1$      ;...AND RESET POINTER.
55 174620 011603      MOV (SP),P3
56 174622 005001      CLR R1
57 174624 005065 170002  CLK BYTES(STOP)      ; CLEAR BYTE COUNTERS.

```

```

58 174630 112714 000200      MOVR  #200,(R4)      ; SFT "MIE"...
59 174634 112764 000224 000002  MOVR  #224,MCC(R4)  ;...AND ENABLE ALL THREE PORTS.
60 174642 116400 000032      MOVB  PADB(R4),R0   ; DUMMY READ (FLUSH)...
61 174646 116400 000034      MOVB  PBDB(R4),R0   ;...BOTH DB'S.
62 174652 106427 000000      MTPS  #PRO "
63                                         ; LOWER CPU...
64 174656 026527 170002 011022 2$:  CMP   BYTES($STOP),#<18.*BITR>118. ; 18 BYTES TRANSFERRED ??
65 174664 001403             BEQ   3$          ; BR IF SO.
66 174666 077105             SOB   R1,2$        ; HANG-AROUND FOR A WHILE.
67 174670 052715 000100      BIS   #BIT6,($EF)  ;; E6 = LOOP TIME-OUT, DATA XFER INCOMPLETE
68
69 174674 012603             3$:  MOV   (SP)+,R3      ; GET RCV'D...
70 174676 012702 163646      MOV   #XFILT10-K3K,R2  ;...AND XMT'D POINTERS...
71 174702 012701 000022      MOV   #18.,R1
72 174706 122223             4$:  CMPB (P2)+,(P3)+  ;...AND COMPARE XMTD/RCVD DATA.
73 174710 001402             BEQ   5$          ;; E7 = RECEIVED DATA INCORRECT
74 174712 052715 000200      BIS   #BIT7,($EF)
75 174716 077105             5$:  SOB   R1,4$        ;; E7 = RECEIVED DATA INCORRECT
76
77 174720 105764 000100      6$:  TSTB  PAMS(R4)    ; CHECK PORT A DIRECTION.
78 174724 100404             BMI   7$          ; BR IF OUTPUT.
79 174726 105014             CLR   (R4)         ; ELSE, CLEAR MASTER...
80 174730 105064 000002      CLR   MCC(R4)     ;...CONTROL REGISTERS...
81 174734 000676             BR    PIO02       ;...AND GO 'ROUND ONCE MORE.
82
83 174736 112714 000001      7$:  MOVB  #1,(R4)      ; THEN, RESET THE CHIP...
84 174742 005064 000140      CLR   XCVR(R4)    ;...AND TRANSCIVER CONTROLS.
85 174746
86 174746 012637 000162      PIOXIT: MOVB  #1,(R4)      ; THEN, RESET THE CHIP...
                                         ;...AND TRANSCIVER CONTROLS.
                                         ; RESTORE PIOBV.
174752 012637 000160      MOVB  (SP)+,PIOBV
174756 012637 000156      MOVB  (SP)+,PIOAV+2 ; RESTORE PIOAV.
174762 012637 000154      MOVB  (SP)+,PIOAV
174766 012637 000106      MOVB  (SP)+,PEVNT+2 ; RESTORE PEVNT.
174772 012637 000104      MOVB  (SP)+,PEVNT
174776 011500               MOV   ($EF),R0   ;; ERROR BITS => R0<11:00>...
175000 001402               BEQ   30007$    ;....AND SKIP IF NONE.
175002 052700 100000      BIS   #<10*BIT12>,R0 ;; ELSE, ADD TEST NUM => R0<15:12>...
175006 000207               30007$: RETURN ;....AND RETURN (NZ).

```

```
1 ;  
2 ; PORT INTERRUPT HANDLER(S).  
3 ;  
4 .ENABLEL LSB  
5 175010 012700 000032 AT: MOV #PADB,R0 ; PORT A -- GET DB INDEX.  
6 175014 105764 000100 TSTB PAMS(R4) ; CHECK DIRECTION.  
7 175020 100407 BMI 1S ; OUTPUT  
8 175022 000421 BR 3$ ; INPUT  
9  
10 175024 012700 000034 BI: MOV #PBDB,R0 ; PORT B -- DITTO.  
11 175030 105764 000120 TSTB PBMS(P4)  
12 175034 100401 BMI 1S ; OUTPUT  
13 175036 000413 BR 3$ ; INPUT  
14  
15 175040 060400 1$: ADD R4,R0 ; OUTPUT -- POINT TO DATA BUFFER.  
16 175042 105712 TSTR (R2) ; ANY BYTES LEFT ??  
17 175044 001404 BEQ 2$ ; BR IF NOT.  
18 175046 112210 MOVB (R2)+,(R0) ; ELSE, SFND NEXT BYTE...  
19 175050 105265 170002 INCB BYTES($TOP) ;...BUMP THE COUNT...  
20 175054 000410 BR 4$ ;...AND RETURN.  
21 175056 112760 000240 177766 2$: MOVB #240,-12(R0) ; WHEN XMIT DONE, CLEAR "IP"...  
22 175064 000404 BR 4$ ;...AND RETURN.  
23  
24 175066 060400 3$: ADD R4,R0 ; INPUT -- POINT TO DATA BUFFER.  
25 175070 111023 MOVB (R0),(R3)+ ; STORE THE BYTE...  
26 175072 105265 170003 INCB BYTES+1($TOP) ;...AND BUMP THE COUNT.  
27 175076 112760 000140 177766 4$: MOVB #140,-12(R0) ; COMMON EXIT -- CLEAR "IUS"...  
28 175104 000002 RTI ;...AND RETURN.  
29 .DISABLEL LSB
```

```

1           .SBTTL - T11 DMA CONTROLLER AMZ8016
2           ;*****
3           ;** TEST 11 -- DMA CONTROLLER AMZ8016
4           ;*****
5           175106
6           175106
7           175106 013746 000164      TST11:
8           175112 013746 000166      TDMA:
9           175116 005015      MOV     DMAV,-(SP)      ; SAVE DMAV.
10          ;           MOV     DMAV+2,-(SP)
11          ;           CLR     ($EF)      ; CLEAR ERROR FLAG.
12          ;
13          ; INPUT: R1 = Q-BUS ADDRESS BITS<21:6> OR ZERO.
14          ; OUTPUT: R0 = ERROR FLAGS OR ZERO.
15          ;
16          ;           T11 DMA CONTROLLER AMZ8016
17          ;           .
18          ;           E0 = ROM IN VECTOR SPACE -- INTERRUPTS NOT TESTED
19          ;           .
20          ;           E1 = Q-BUS ADDRESS UNDEFINED -- Q-BUS ACCESS NOT TESTED
21          ;           .
22          ;           E2 = CHAN INTERRUPT NOT RECEIVED
23          ;           .
24          ;           E3 = DMA CHANNEL HUNG (TCIEOP BOTH CLEAR)
25          ;           .
26          ;           E4 = DMA ABORTED (EOP = 1 = NXM)
27          ;           .
28          ;           E5 = DMA DATA ERROR
29          ;
30          170002      QAHI= TMP1      ; SAVERS FOR A TWO WORD Q-BUS ADDRESS...
31          170004      QALO= TMP2
32          170006      CLISP= TMP3      ;...THREE VARIABLE (CH1/CH2) COMMANDS...
33          170010      SETIE= TMP4
34          170012      STACHN= TMP5
35          ;
36          175120 004737 163602      CALL    XRAMLO-K3K      ; VECTOR SPACE USABLE ??
37          175124 001402      BEQ    1$      ; SKIP IF SO.
38          175126 052715 000001      BIS    #BIT0,($EF)      ; E0 = ROM IN VECTOR SPACE -- INTERRUPTS NOT TESTED
39          ;
40          175132 010102      1$:      MOV    R1,R2      ; Q-ADDR<21:6> DEFINED ??
41          175134 001002      BNE    2$      ; SKIP IF SO.
42          175136 052715 000002      BIS    #BIT1,($EF)      ; E1 = Q-BUS ADDRESS UNDEFINED -- Q-BUS ACCESS NOT TESTED
43          ;
44          175142 012700 000006      2$:      MOV    #6,R0
45          175146 005001      CLR    R1
46          175150 006302      3$:      ASL    R2      ; \
47          175152 006101      ROL    R1      ; > ADDR<21:16> => R1<6:0>
48          175154 077003      SUB    R0,3$      ; /
49          175156 000301      SWAB   R1      ; REPOSITION TO DMA FORMAT <13:8>...
50          175160 052701 100000      BIS    #BIT15,R1      ;...SET "Q-BUS" BIT...
51          175164 010165 170002      MOV    R1,QAHI($TOP)      ;...AND SAVE Q-BUS HIGH (SEG/TAG)...
52          175170 010265 170004      MOV    R2,QALO($TOP)      ;...AND LOW (OFFSET) ADDRESS.
53          ;
54          ; NOW PUSH THE 6-LINK CHAIN SCRIPT ONTO THE STACK
55          ; FILLING IN THE CORRECTED ADDRESSES AS WE GO.
56          ;
57          175174 012700 162102      TDMA1: MOV    #CHAINX+2-K3K,R0      ; CHAIN (TOP) ADDRESS...
58          175200 012701 000052      MOV    #<CHAINX+2-CHAIN1>/2,R1 ;...AND SIZE.
59          175204 014046      3$:      MOV    -(R0),-(SP)      ; PUSH A WORD.
60          175206 021627 150002      CMP    (SP),#BUFR1      ; LOCAL BUFR1 ??
61          175212 001002      BNE    4$      ; NO.
62          175214 060516      ADD    STOP,(SP)      ; YES, ADJUST IT.
63          175216 000426      BR    8$      ;
64          175220 021627 160002      4$:      CMP    (SP),#BUFR2      ; LOCAL BUFR2 ??
65          175224 001002      HNE    5$      ; NO.

```

```

50 175226 060516          ADD    $STOP,(SP)      ; YES, ADJUST IT.
51 175230 000421          BR     8$                ;
52 175232 021627 120000    5$:   CMP    (SP),#120000 ; Q-BUS 20K ??
53 175236 001003          BNE    6$                ; NO.
54 175240 016516 170004    MOV    QALO($STOP),(SP) ; YES, SUBSTITUTE IT.
55 175244 000407          BR     7$                ;
56 175246 021627 130000    6$:   CMP    (SP),#130000 ; Q-BUS 22K ??
57 175252 001010          BNE    8$                ; NO.
58 175254 016516 170004    MOV    QALO($STOP),(SP) ; YES, SUBSTITUTE IT...
59 175260 062716 010000    ADD    #4096,,(SP)   ;...AND BUMP UP 2KW.
60 175264 014046          7$:   MOV    -(R0),-(SP)  ; ON EITHER Q-BUS ADDRESS, PUSH...
61 175266 016516 170002    MOV    QAHI($STOP),(SP) ;...AND SUBSTITUTE THE HIGH PART...
62 175272 005301          DEC    R1                ;...AND TWEAK THE COUNT ACCORDINGLY.
63 175274 077135          8$:   SOB    R1,3$      ; LOOP 'TIL DONE...
64                               ;...(SP) IS THE CHIP'S CHAIN ADDRESS.

65                               ;
66                               ; NOW, EXECUTE AND ERROR CHECK 5 DMA CHAINS AS FOLLOWS:
67                               ; LOCAL => LOCAL, 2KW.
68                               ; LOCAL I/O PAGE => LOCAL, 1KW.
69                               ; LOCAL => Q-BUS 20K, 2KW.
70                               ; Q-BUS 20K => Q-BUS 22K, 2KW.
71                               ; Q-BUS 22K => LOCAL, 2KW.

72                               ;
73 000070          MMR= 70           ; MASTER MODE REGISTER.
74 000056          STAT1= 56         ; CHANNEL 1 STATUS.
75 000054          STAT2= 54         ; CHANNEL 2 STATUS...
76 000054          CMDR= 54         ;...AND COMMAND REGISTER (FOR BOTH).
77 000046          CHA1H= 46        ; CHAIN ADDRESS REGISTERS HIGH (SEG/TAG)...
78 000044          CHA2H= 44        ;
79 000042          CHA1L= 42        ;...AND THE LOW (OFFSET) HALFS.
80 000040          CHA2L= 40        ;
81 000001          TC= 1            ; TERMINAL COUNT STATUS BIT (BIT0).
82 000002          EOP= 2            ; END-OF-PROCESS (NFM) STATUS BIT (BIT2).
83 020000          IP= 20000        ; INTERRUPT POSTED (DONE) STATUS BIT (BIT13).

84                               ;
85 175276 106427 000200    TDMA2: MTPS  #PR4       ; RAISE TO CHIP LEVEL.
86 175302 012700 174400    MOV   #SDMA,R0      ; BASE ADDRESS.
87 175306 012701 000056    MOV   #46,,R1      ;
88 175312 005060 000054    CLR   CMDR(R0)    ; COMMAND "RESET"...
89 175316 005020          1$:   CLR   (R0)+      ;...AND CLEAR ALL REGISTERS.
90 175320 077102          SOB   R1,1$      ;
91 175322 012701 150002    MOV   #BUFR1,R1    ;
92 175326 060501          ADD   STOP,R1      ; GET BUFR1 ADDRESS.
93 175330 012702 004000    MOV   #2048,,R2    ;
94 175334 010100          2$:   MOV   R1,R0      ;
95 175336 005061 010000    CLR   4096,(R1)   ; CLEAR BUFR2...
96 175342 010021          MOV   R0,(R1)+    ;...AND PUT ADDRESSES IN BUFR1.
97 175344 077205          SOB   R2,2$      ;

98                               ;
99 175346 012704 174400    TDMA3: MOV   #SDMA,R4    ; GET BASE POINTER.
100 175352 012737 161500 000164   MOV   #21S-K3K,DMAV ; INT SERVICE CHAN 1 (AND 2).
101 175360 012737 000200 000166   MOV   #PR4,DMAV+2  ;
102 175366 012765 000054 170006   MOV   #40114,CLISP(STOP) ; SET "CLEAR IUSIIP"...
103 175374 012765 000062 170010   MOV   #42120,SETIE(STOP) ;...SET IE...
104 175402 012765 000240 170012   MOV   #240,STACHN(STOP) ;...AND "START CHAIN" COMMANDS FOR CH 1.
105 175410 012764 000115 000070   MOV   #115,MMR(R4)   ; SET MODE = VI, WAIT, CPINTLV, ENABLE.
106 175416 010664 000042          MOV   SP,CHA1L(R4)  ; SET CHAIN ADDRESS(ES)...

```

```

    107 175422 010664 000040      MOV    SP,CHA2L(R4) ;... = 1ST LINK.
    108 175426 012703 161742      1$:    MOV    #WCTBL-K3K,R3 ; SET WC TABLE POINTER IN R3.
    109 175432 005000              2$:    CLR    R0 ; SET A KEEP-ALIVE TIMER.
    110 175434 032715 000001      BIT    #BIT0,($EF) ; INTERRUPTS USEABLE ???
    111 175440 001001              BNE    2$ ; SKIP NEXT IF NOT.
    112 175442 106400              MTPS   R0 ; YES, LOWER CPU.
    113 175444 016564 170010 000054 25$:  MOV    SRTIE(STOP),CMDR(R4) ; SET "CIE"...
    114 175452 016564 170012 000054      MOV    STACHN(STOP),CMDR(R4) ;...AND START CHAIN CHAN 1(2).
    115 175460 077001              SOB    R0,. ; TC (OR EOP) INTERRUPT SHOULD...
    116                                ;...HAPPEN BEFORE THIS TIMES OUT.
    117 175462 052715 000004      BIS    #BIT2,($EF) ; E2 = CHAN INTERRUPT NOT RECEIVED
    118 175466 024646              CMP    -(SP),-(SP) ; FAKE IT...
    119 175470 026527 170012 000240      CMP    STACHN(STOP),#240 ;...AND SERVICE CHAN 1...
    120 175476 001003              BNE    2$ ;...OR CHAN 2.

    121                                ; PROCESS DMA CHANNEL INTERRUPTS.
    122
    123
    124 175500 016400 000056      21$:  MOV    STAT1(R4),R0 ; GET CHAN 1 STATUS.
    125 175504 000402              BR    27$ ; OR
    126 175506 016400 000054      22$:  MOV    STAT2(R4),R0 ; GET CHAN 2 STATUS.
    127 175512 022626              27$:  CMP    (SP)+,(SP)+ ; ADJUST STACK.
    128 175514 032700 000001      BIT    #TC,R0
    129 175520 001010              BNE    24$ ; BR IF TC IS SET.
    130 175522 032700 000002      BIT    #EOP,R0
    131 175526 001003              BNE    23$ ; BR IF EOP (NMW) IS SET.
    132 175530 052715 000010      BTS    #BIT3,($EF) ; E3 = DMA CHANNEL HUNG (TC|EOP BOTH CLEAR)
    133 175534 000402              BR    24$ ; E4 = DMA ABORTED (EOP = 1 = NMW)
    134 175536 052715 000020      23$:  BIS    #BIT4,($EF) ; CLISP(STOP),CMDR(R4) ; INT SERVICE COMPLETE, CLEAR "IUSIIP".
    135 175542 016564 170006 000054 24$:  MOV    (R3)+,R0 ; GET WORD COUNT FROM "WCTBL"...
    136
    137 175550 012300              3$:    BEQ    6$ ;...AND SKIP IF DATA CHECK INHIBITED.
    138 175552 001423              MOV    #BUFR1,R1
    139 175554 012701 150002      ADD    STOP,R1 ; ASSUME SRC WAS BUFR1...
    140 175560 060501              MOV    #BUFR2,R2 ;...AND DEST BUFR2.
    141 175562 012702 160002      ADD    STOP,R2
    142 175566 060502              CMP    R0,#2048. ; WC = 2KW ???
    143 175570 020027 004000      BEQ    4$ ; SKIP IF SO...
    144 175574 001404              MOV    #164000,R1 ;...ELSE, SRC WAS ODTROM...
    145 175576 012701 164000      SUB    #4096.,R2 ;...AND DST BUFR1.
    146 175602 162702 010000      CMP    (R1)+,(R2)+ ; COMPARE SRC/DST DATA.
    147 175606 022122              BEQ    5$ ; BR IF OK.
    148 175610 001403              BTS    #BIT5,($EF) ; E5 = DMA DATA ERROR
    149 175612 052715 000040      BR    6$ ; ESCAPE ON 1ST DATA ERROR.
    150 175616 000401              5$:    SOB    R0,4$ ; 152
    151 175620 077006              6$:    TST    (R3) ; NEXT LINK INVOLVE Q-BUS ???
    152
    153 175622 005713              BNE    7$ ; BR IF NOT.
    154 175624 001003              BIT    #BIT1,($EF) ; YES, Q-BUS ACCESS PERMITTED ???
    155 175626 032715 000002      BNE    8$ ; NO, Q-BUS ACCESS PERMITTED ???
    156 175632 001003              CMP    (R3),#-1 ; CHECK FOR END OF TABLE...
    157 175634 021327 177777      BNE    2$ ;...AND LOOP 'TIL ALL LINKS DONE.
    158 175640 001274              159
    160 175642 032765 000001 170012 8$:  BIT    #1,STACHN(STOP) ; NOW, CHECK THE CHANNEL SELECT BIT...
    161 175650 001012              BNE    9$ ;...AND EXIT IF BOTH DONE.
    162 175652 062737 000006 000164      ADD    #22$-21$,DMAV ; ELSE, CHANGE VECTOR...

```

```

163 175660 005265 170006           INC CLISP($TOP)
164 175664 005265 170010           INC SETIE($TOP) ;...AND COMMANDS FOR CHAN 2...
165 175670 005265 170012           INC STACHN($TOP)
166 175674 000654                 BR 1$      ;...AND GO 'ROUND ONCE MORE.
167
168 175676 005064 000054           9$: CLR CMDR(R4) ; ALL DONE RESET CHIP.
169 175702 062706 000124           ADD #CHAINX+2-CHAIN1,SP ; POP CHAIN OFF THE STACK.
170 175706 032715 000001           BIT #BIT0,(SEF) ; DID WE RUN WITH INTERRUPTS ???
171 175712 001402                 BEQ 10$    ; SKIP IF SO.
172 175714 042715 000004           BIC #BIT2,(SEF) ; NO, DISCARD INTERRUPT ERROR BIT.
173 175720
175720 012637 000166           10$:DMAXIT: MOV (SP)+,DMAV+2 ; RESTORE DMAV.
175724 012637 000164           MOV (SP)+,DMAV
175730 011500                 MOV (SEF),R0 ; ERROR BITS => R0<11:00>...
175732 001402                 BEQ 30008$ ;...AND SKIP IF NONE.
175734 052700 110000           BIS #<11*BIT12>,R0 ; ELSE, ADD TEST NUM => R0<15:12>...
175740 000207                 30008$: RETURN ;....AND RETURN (NZ).

174
175 ; WORD COUNT TABLE FOR POST-DMA DATA CHECKS.
176 ; FINAL CHECK (5.) VERIFIES THE RESULTS OF CHAINS 2, 3, 4, AND 5.
177 ;
178 175742 004000           WCTBL: 2048. ; 1. 2KW, BUFR1 => BUFR2.
179 175744 002000           1024. ; 2. 1KW, ODTROM => BUFR1.
180 175746 000000           0 ; 3. Q-BUS INVOLVED, NO DATA CHECK.
181 175750 000000           0 ; 4. Q-BUS INVOLVED, NO DATA CHECK.
182 175752 004000           2048. ; 5. 2KW, BUFR1 => Q1 => Q2 => BUFR2.
183 175754 177777           -1 ; TABLE TERMINATOR.

184
185 ; AND THIS IS THE 6 LINK DMA CHAIN.
186 ;
187 ; THIS GETS PUSHED ONTO THE STACK WHERE LOCAL AND Q-BUS ADDRESSES
188 ; ARE ADJUSTED ACCORDING TO RAM CONFIG AND ENTRY OPTION (R2).
189 ;
190 175756 001606           CHAIN1: 1606 ; LOAD ARA, ARB, OPK, VECT, AND CHAN MODE.
191 175760 000000 150002          0, BUFR1 ; FROM LOCAL BUFR1...
192 175764 000000 160002          0, BUFR2 ;...TO LOCAL BUFR2...
193 175770 004000           2048. ;...2K WORDS.
194 175772 000164           DMAV ; VECTOR.
195 175774 000030 001340          30, 1340 ; SFTREQ, HDMSK, INT TCIEOP, INTLV, WORD-WORD.

196
197 176000 001602           CHAIN2: 1602 ; RE-LOAD ARA, ARB, OPK, AND CHAN MODE ONLY.
198 176002 000000 164000          0, 164000 ; FROM LOCAL ODTROM...
199 176006 000000 150002          0, BUFR1 ;...TO LOCAL BUFR1...
200 176012 002000           1024. ;...1K WORDS...
201 176014 000030 001340          30, 1340 ;...SAME MODE AND TERMINATION.

202
203 ;Previously, CHAIN3 read the arbiter's boot ROM using the QBUS. However, the
204 ;Boot ROM of a KDP11-8 does not respond on the QBUS. Therefore, CHAIN3 was
205 ;removed.
206
207 176020 001602           CHAIN4: 1602 ; DITTO.
208 176022 000000 150002          0, BUFR1 ; FROM LOCAL BUFR1...
209 176026 100000 120000          100000, 120000 ;...TO Q-BUS 20K...
210 176032 004000           2048. ;...2K WORDS.
211 176034 000030 001340          30, 1340 ; DITTO.

212
213 176040 001602           CHAIN5: 1602 ; DITTO.

```

KXTMP - KXT11CA Native Firmware MACRO V05.00 00:51 Page 113-4
- T11 DMA CONTROLLER AM28016

214 176042 100000 120000	100000, 120000 ; FROM Q-BUS 20K...
215 176046 100000 130000	100000, 130000 ;...TO Q-BUS 22K...
216 176052 004000	2048. ;...2K WORDS.
217 176054 000030 001340	30, 1340 ; DITTO.
218	
219 176060 001602	CHAIN6: 1602 ; DITTO.
220 176062 100000 130000	100000, 130000 ; FROM Q-BUS 22K...
221 176066 000000 160002	0, BUFR2 ;...TO LOCAL BUFR2...
222 176072 004000	2048. ;...2K WORDS.
223 176074 000030 001340	30, 1340 ; DITTO.
224 176100 000000	CHAINX: 0 ; TERMINATOR.

145

```

1           .SBTTL - T12  Q-BUS INTERRUPT
2           ;*****
3           ;** TEST 12 -- Q-BUS INTERRUPT
4           ;*****
5           176102          TST12:
6           176102          TQIR:
7           176102 013746 000130      MOV    BIACKV,-(SP)   ; SAVE BIACKV.
8           176106 013746 000132      MOV    BIACKV+2,-(SP)
9           176112 013746 000024      MOV    PWRV,-(SP)    ; SAVE PWRV.
10          176116 013746 000026      MOV    PWRV+2,-(SP)
11          176122 005015          CLR    ($EF)        ; CLEAR ERROR FLAG.

12          ; INPUT: R1 = 1ST OF 2 CONSECUTIVE Q-BUS VECTORS.
13          ; OUTPUT: R0 = ERROR FLAGS OR ZERO.
14          ;
15          ; *** REQUIRES A RUNNING ARBITER CPU ***
16          ;
17          ; VERIFY THAT THE QIR CAN INTERRUPT THE ARBITER, AND THAT
18          ; THE POST-INTERRUPT PROCESS WORKS AS ADVERTISED.
19          ;
20          176124 005701          TST    R1          ; Q VECTOR DEFINED ??
21          176126 001003          BNE    1$          ; PROCEED IF SO.
22          176130 052715 000001          BIS    #BIT0,$(EF)  ;; EO = Q-BUS VECTOR UNDEFINED -- CAN'T RUN
23          176134 000516          BR     QIXIT
24          ;
25          176136 106427 000240          MTPS   #PR5        ; RAISE CPU.
26          176142 042737 177677 177530 1$:    BIC    #*CBIT6,SCSR  ; ENSURE EVERYTHING OFF (EXCEPT DPR).
27          176150 010137 177532          MOV    R1,SQTR    ; WRITE A VECTOR IN QIR<9:2>.
28          176154 032737 040000 177530  BIT    #BIT14,SCSR  ; REQUEST SHOULD BE SET.
29          176162 001002          BNE    2$          ; SKIP IF SO.
30          176164 052715 000002          BIS    #BIT1,$(EF)  ;; E1 = BREQ (CSR<14>) NEVER GOT SET
31          ;
32          176170 004737 163602          2$:    CALL   XRAMLO-K3K  ; VECTOR SPACE USABLE ??
33          176174 001403          BEQ    3$          ; PROCEED IF SO.
34          176176 052715 000004          BIS    #BIT2,$(EF)  ;; E2 = ROM IN VECTOR SPACE -- RESPONSE INTERRUPTS NOT TESTE
35          176202 000473          BR     QIXIT
36          ;
37          176204 012737 162232 000130 3$:    MOV    #4$-K3K,BIACKV ; SET RESPONSE VECTOR.
38          176212 012737 000240 000132          MOV    #PR5,BIACKV+2
39          176220 052737 030000 177530          BTS    #BIT13#BIT12,SCSR ; SET ENABLES.
40          176226 077001          SOB    R0,.        ; RESPONSE SHOULD BE MASKED AT THIS LEVEL.
41          176230 000403          BR     5$          ;
42          176232          4$:    BIS    #BIT3,$(EF)  ;; E3 = INT-ON-BIACK NOT MASKED AT PR5
43          176236 000411          BR     6$          ;
44          176240 062737 000030 000130 5$:    ADD    #6$-4$,BIACKV ; CHANGE RESPONSE VECTOR.
45          176246 106427 000000          MTPS   #PRO        ; LOWER CPU...
46          176252 077001          SOB    R0,.        ;...INT SHOULD COME IN.

```

```

    47 176254 052715 000020      BIS     #BIT4,$(SEF)    ;; F4 = INT-ON-BIACK NOT RECEIVED
    48 176260 000401      BR      65$                ;;
    49 176262 022626      6S:    CMP     $(SP)+,(SP)+    ; OK, FIX STACK.
    50 176264 032737 040000 177530 65$:    BIT     #BIT14,SCSR    ; REQUEST SHOULD BE CLEAR.
    51 176272 001405      BEQ     7$                ;;
    52 176274 052715 000040      BIS     #BIT5,$(SEF)    ;; F5 = BIACK DIDN'T CLEAR BREQ (CSR<14>)
    53 176300 077001      SOB     R0,.              ;;
    176302 000005      RESET   R0,.              ;; LOCAL RESET.
    176304 077001      SOB     R0,.              ;;

    54      ;;
    55      ; NOW ONCE MORE USING THE ALTERNATE VECTOR (IN ARBITEP)
    56      ; TO CAUSE A "BUS-INIT" AND LOCAL TRAP THRU VECTOR 24.
    57      ;
    58 176306 022121      7S:    CMP     (R1)+,(R1)+    ; BUMP UP (+4) TO SECOND VECTOR.
    59 176310 012737 162370 000024      MOV     #8S-K3K,PWRV    ; SET PWK-UP VECTOR.
    60 176316 012737 000340 000026      MOV     #PR7,PWRV+2    ;;
    61 176324 106427 000340      MTPS   #PR7            ; SHOULD BE NON-MASKABLE.
    62 176330 042737 020000 177530      BIC     #BIT13,$CSR    ; CLEAR JACK ENABLE.
    63 176336 052737 002000 177530      BIS     #BIT10,$CSR    ; ARM BRESET TRAP.
    64 176344 010137 177532      MOV    R1,SQIR        ; POST Q-BUS REQUEST.
    65 176350 005000      CLR    R0               ; ARBITER EXECUTES A BUS-RESET...
    66 176352 077001      SOB    R0,.            ;...AND A TRAP THRU 24 SHOULD OCCUR.
    67 176354 052715 000100      BIS     #BIT6,$(SEF)    ;; F6 = BRESET TRAP THRU 24 DIDN'T HAPPEN
    68 176360 077001      SOB    R0,.            ;;
    176362 000005      RESET   R0,.            ;; LOCAL RESET.
    176364 077001      SOB    R0,.            ;;
    69 176366 000401      dR     QIXIT          ;;
    70 176370 022626      8S:    CMP     (SP)+,(SP)+    ; PWR-UP TRAPPED TO HERE, FIX STACK.
    71 176372 042737 177677 177530  QIXIT:  BTC     #^CBIT6,SCSR    ; CLEAR CSR.
    72 176400 012637 000026      MOV     (SP)+,PWRV+2    ;; RESTORE PWRV.
    176404 012637 000024      MOV     (SP)+,PWRV        ;;
    176410 012637 000132      MOV     (SP)+,BIACKV+2    ;; RESTORE BIACKV.
    176414 012637 000130      MOV     (SP)+,BIACKV        ;;
    176420 011500      MOV    $(SEF),R0        ;; ERROR BITS => R0<11:00>...
    176422 001402      BEQ     30009$          ;;...AND SKIP IF NONE.
    176424 052700 120000      BIS     #<12*BIT12>,R0    ;; ELSE, ADD TEST NUM => R0<15:12>...
    176430 000207      30009$: RRETURN      ;;...AND RETURN (NZ).

```

14

```

1 SBTTL - T13 DUAL-PORT RAM AND IPV
2 ;***** TEST 13 -- DUAL-PORT RAM AND IPV *****
3 ;***** TEST 13 -- DUAL-PORT RAM AND IPV *****
4 ;***** TEST 13 -- DUAL-PORT RAM AND IPV *****
5 ;***** TEST 13 -- DUAL-PORT RAM AND IPV *****
6 ;***** TEST 13 -- DUAL-PORT RAM AND IPV *****
7 ;***** TEST 13 -- DUAL-PORT RAM AND IPV *****
8 ;***** TEST 13 -- DUAL-PORT RAM AND IPV *****
9 ;***** TEST 13 -- DUAL-PORT RAM AND IPV *****
10 ;***** TEST 13 -- DUAL-PORT RAM AND IPV *****
11 ;***** TEST 13 -- DUAL-PORT RAM AND IPV *****
12 ;***** TEST 13 -- DUAL-PORT RAM AND IPV *****
13 ;***** TEST 13 -- DUAL-PORT RAM AND IPV *****
14 ;***** TEST 13 -- DUAL-PORT RAM AND IPV *****
15 ;***** TEST 13 -- DUAL-PORT RAM AND IPV *****
16 ;***** TEST 13 -- DUAL-PORT RAM AND IPV *****
17 ;***** TEST 13 -- DUAL-PORT RAM AND IPV *****
18 ;***** TEST 13 -- DUAL-PORT RAM AND IPV *****
19 ;***** TEST 13 -- DUAL-PORT RAM AND IPV *****
20 ;***** TEST 13 -- DUAL-PORT RAM AND IPV *****
21 ;***** TEST 13 -- DUAL-PORT RAM AND IPV *****
22 ;***** TEST 13 -- DUAL-PORT RAM AND IPV *****
23 176432 013746 000120
24 176432 013746 000122
25 176442 013746 000124
26 176446 013746 000126
27 176452 013746 000134
28 176456 013746 000136
29 176462 005015
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
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
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
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
789
789
790
791
792
793
794
795
796
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
889
889
890
891
892
893
894
895
896
897
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
988
989
989
990
991
992
993
994
995
996
997
998
999

```

```

    46 176532 062700 160000      ADD    #$QDPR1, R0 ;...PLUS LO BASE = GLOBAL ADDRESS.
    47 176536 032737 000200 177524      BIT    #BIT7,$UID ; ID >= 8. ???
    48 176544 001402      BEQ    1S ; SKIP IF NOT.
    49 176546 062700 015400      ADD    #<$QDPR2-$QDPR1>,R0 ; YES, ADJUST TO HIGH RANGE.
    50 176552 010065 170004      1$:   MOV    R0,QADR($STOP) ; SAVE IT IN RAM.
    51 176556 010500      MOV    STOP, R0
    52 176560 062700 150002      ADD    #BUFIR1, R0 ; CALCULATE BUFFER ADDRESS...
    53 176564 010065 170006      MOV    R0,LADR($STOP) ;...AND SAVE IT AS WELL.
    54 176570 012700 163602      MOV    #QIOC+2-K3K,R0 ; GET DRIVER ADDRESS...
    55 176574 012701 000101      MOV    #<QIOC+2-RQIO>/2,R1 ;...AND SIZE...
    56 176600 014046      2$:   MOV    -(R0),-(SP) ;...AND PUSH IT ONTO THE STACK.
    57 176602 077102      SOB    R1,2$ ;...AND PUSH IT ONTO THE STACK.
    58 176604 000403      BR    35$ ; SKIP NEXT.
    59 176606 052765 002000 170004 3$:   BIS    #2000,QADR($STOP) ; ALTERNATE BASE ADDRESS.
    60 176614 016500 170004 35$:   MOV    QADR($STOP), R0 ; SET Q-BUS ADDRESS...
    61 176620 016501 170006      MOV    LADR($STOP), R1 ;...LOCAL ADDRESS...
    62 176624 012702 000020      MOV    #16.,R2 ;...AND WORD COUNT.
    63 176630 004766 000000      CALL   RD(SP) ; READ FROM Q-BUS SIDE VIA DMA.
    64 176634 102007      BVC    4$ ; BR IF OK (NOT "EOP").
    65 176636 032765 002000 170004      BIT    #2000,QADR($STOP) ; NOT THERE, BOTH BASES TRIED ???
    66 176644 001760      BEQ    3$ ; NO, GO 'ROUND ONCE.
    67 176646 052715 000002      BIS    #BIT1,(SEF) ; E1 = BUS-ERROR ON Q-BUS DPR READ
    68 176652 000411      BR    TIPV1
    69 176654 016500 170006      4$:   MOV    LADR($STOP), R0 ; GET BUFFER ADDRESS.
    70 176660 012701 000020      MOV    #16.,R1
    71 176664 005720      5$:   TST    (R0)+ ; OBSERVED DATA SHOULD BE ZERO.
    72 176666 001402      BEQ    6$ ; BR IF SO.
    73 176670 052715 000004      BIS    #BIT2,(SEF) ; E2 = NON-ZERO DATA RETURNED ON Q-BUS READ
    74 176674 077105      6$:   SOB    R1,5$ ;...AND PUSH IT ONTO THE STACK.
    75
    76 ; SO FAR, SO GOOD -- NOW WRITE ALL FROM THE Q-BUS SIDE.
    77 ; 1ST WRITE SHOULD FORCE A NON-MASKABLE RESTART TRAP (COMMAND).
    78 ; ALL OTHERS SHOULD APPEAR NON-EXISTANT (EOP).. .
    79
    80 176676 004737 163602      TIPV1: CALL   XRAMLO-K3K ; VECTOR SPACE USABLE ???
    81 176702 001403      BEQ    1S ; PROCEED IF SO.
    82 176704 052715 000010      BIS    #BIT3,(SEF) ; E3 = ROW IN VECTOR SPACE -- Q-WRITE AND INTERRUPT NOT TES
    83 176710 000563      BR    TIPXIT
    84
    85 176712 012700 175000      1$:   MOV    #SDPR,R0
    86 176716 016501 170006      MOV    LADR($STOP), R1
    87 176722 012702 000020      MOV    #16.,R2
    88 176726 005020      11$:  CLR    (R0)+ ; CLEAR DPR...
    89 176730 005021      CLR    (R1)+ ;...AND BUFFER.
    90 176732 077203      SOB    R2,11$ ;...AND PUSH IT ONTO THE STACK.
    91
    92 176734 012701 000120      MOV    #DPRV4,R1
    93 176740 012721 163350      MOV    #IPV4-K3K,(R1)+ ; SET DPRV4...
    94 176744 012721 000240      MOV    #PR5,(R1)+ ;...AND DPRV5...
    95 176750 012721 163356      MOV    #IPV8-K3K,(R1)+ ;...AND DPRV8...
    96 176754 012721 000240      MOV    #PR5,(R1)+ ;...AND DPRV5...
    97 176760 022121      CMP    (R1)+,(R1)+ ;...AND DPRV12 VECTORS.
    98 176762 012721 163364      MOV    #IPV12-K3K,(R1)+ ;...AND DPRV12 VECTORS.
    99 176766 012721 000240      MOV    #PR5,(R1)+ ;...AND DPRV5...
    100 176772 106427 000340     MTPS   #PR7 ; RAISE CPU, COMMAND INTERRUPT...
    101                                ;...SHOULD BE NON-MASKABLE.
    102 176776 016500 170004      MOV    QADR($STOP), R0 ; GET GLOBAL ADDRESS...

```

```

103 177002 016501 170006      MOV   LADR($TOP),R1 ;...LOCAL ADDRESS...
104 177006 012702 000001      MOV   #1,R2 ;...AND WORD COUNT.
105 177012 012711 000040      MOV   #40,(R1) ; SFT 1ST WORD = "NOP" COMMAND.
106 177016 004766 000010      CALL  WRT(SP) ; WRITE DPR WORD 0...
107 177022 102002             BVC   2$ ;...SHOULD NOT HAVE ABORTED.
108 177024 052715 000020      BIS   #BIT4,(SEF) ;; E4 = BUS ERROR ON WRITE TO DPR WORD 0
109 177030 005737 175000      2$:   TST   SDPR ; COMMAND SHOULD HAVE BEEN CLEARED.
110 177034 001402             BEQ   3$ ;...NO INTERRUPT.
111 177036 052715 000040      BIS   #BIT5,(SEF) ;; E5 = COMMAND INTERRUPT DIDN'T HAPPEN (OR DOESN'T WORK)
112 177042                   3$:   ;...NO INTERRUPT.
113 177042 005004             4$:   CLR   R4 ; CLEAR INT RECEIVED FLAG...
114 177044 106427 000000      MTPS  #PRO ;...AND LOWER CPU.
115 177050 012703 000017      MOV   #15,,R3 ; SET LOOP FOR 15 MORE WRITES.
116 177054 012702 000001      MOV   #1,R2 ; SET WORD COUNT.
117 177060 004766 000010      CALL  WRT(SP) ; WRITE NEXT FROM Q-BUS SIDE...
118 177064 102002             BVC   6$ ;...SHOULD GET "EOP"...
119 177066 005704             TST   R4 ;... AND ...
120 177070 001402             BEQ   7$ ;...NO INTERRUPT.
121 177072                   6$:   BIS   #BIT6,(SEF) ;; E6 = Q-BUS WRITE AND/OR INTERRUPT WITH DPR DISABLED
122 177076 077312             7$:   SOB   R3,SS ;...NO INTERRUPT.
123 ; NOW, ENABLE THE DPR, AND WRITE WORDS 1-15 AGAIN.
124 ; EXPECT LEVEL 5 INTERRUPTS AT WORDS 4, 8, AND 12,
125 ; AND EOP AT WORDS 1, 5, AND 9.
126 ;
127 ;
128 177100 052737 000170 177530 TIPV2: BIS   #170,$CSR ; SET ALL ENABLES.
129 177106 016500 170004      MOV   QADR($TOP),R0 ; SET Q ADDRESS...
130 177112 062700 000002      ADD   #2,R0 ;...FOR DPR1...
131 177116 016501 170006      MOV   LADR(STOP),R1 ;...AND LOCAL ADDRESS.
132 177122 062701 000002      ADD   #2,R1 ;...DITTO.
133 177126 012703 000002      MOV   #BIT1,R3 ; BIT = CURRENT WORD UNDER TEST.
134 177132 005004             1$:   CLR   R4 ; CLEAR "INT RECD" FLAG.
135 177134 012702 000001      MOV   #1,R2 ; SET WC...
136 177140 004766 000010      CALL  WRT(SP) ;...AND WRITE 1ST WORD.
137 177144 102406             BVS   3$ ; BR IF IT TIMED-OUT.
138 ;
139 177146 030327 001042      2$:   BIT   R3,#BIT1#BIT5#BIT9 ; DIDN'T TIME-OUT, SHOULD IT HAVE ???
140 177152 001410             BEQ   4$ ; NO, PROCEED.
141 177154 052715 000200      BIS   #BIT7,(SEF) ;; E7 = Q-BUS WRITE DIDN'T TIME-OUT
142 177160 000405             BR    4$ ;...NO, PROCEED.
143 177162 030327 001042      3$:   BIT   R3,#BIT1#BIT5#BIT9 ; DID TIME-OUT, SHOULD IT HAVE ???
144 177166 001002             BNE   4$ ; YES, PROCEED.
145 177170 052715 000400      BIS   #BIT8,(SEF) ;; E8 = Q-BUS WRITE TIMED-OUT
146 ;
147 177174 030327 000020      4$:   BIT   R3,#BIT4 ; INTFRRUPT 4 EXPECTED ???
148 177200 001405             BEQ   5$ ; NO
149 177202 020427 000004      CMP   R4,#4. ; YES, RECEIVED ???
150 177206 001402             BEQ   5$ ; YES.
151 177210 052715 001000      BIS   #BIT9,(SEF) ;; E9 = DPR INTERRUPT 4 NOT RECEIVED
152 177214 030327 000400      5$:   BIT   R3,#BIT8 ; INTERRUPT 8 EXPECTED ???
153 177220 001405             BEQ   6$ ; NO.
154 177222 020427 000010      CMP   R4,#8. ; YES, RECEIVED ???
155 177226 001402             BEQ   6$ ; YES.
156 177230 052715 002000      BIS   #BIT10,(SEF) ;; E10 = DPR INTERRUPT 8 NOT RECEIVED
157 177234 030327 010000      6$:   BIT   R3,#BIT12 ; INTERRUPT 12 EXPECTED ???
158 177240 001405             BEQ   7$ ; NO

```

```

159 177242 020427 000014      CMP    R4,#12.        ; YES, RECEIVED ???
160 177246 001402      BEQ    7$              ; YES.
161 177250 052715 004000      BIS    #BIT11,(SEF)   ; E11 = DPR INTERRUPT 12 NOT RECEIVED
162 177254 006303      7$:    ASL    R3              ; SHIFT UP TO NEXT WORD (BIT)...
163 177256 001325      BNE    1$              ;...AND LOOP 'TIL ALL DONE.
164
165 177260 062706 000202      TIPXIT: ADD  #<QIOC+2-RQIO>,SP ; DONE, POP DRIVER OFF THE STACK.
166 177264 042737 177677 177530      BIC    #CBIT6,SCSR   ; CLEAR ALL EXCEPT "DPREN".
167 177272 012700 175000      MOV    #$DPR,R0     ; GET LOCAL ADDRESS...
168 177276 012701 000020      MOV    #16.,R1
169 177302 005020      1$:    CLR    (R0)+        ;...AND FLUSH ANY CRAP.
170 177304 077102      SOB    R1,1$          ; SOB
171 177306 012637 000136      MOV    (SP)+,DPRV12+2 ; RESTORE DPRV12.
177312 012637 000134      MOV    (SP)+,DPRV12
177316 012637 000126      MOV    (SP)+,DPRV8+2 ; RESTORE DPRV8.
177322 012637 000124      MOV    (SP)+,DPRV8
177326 012637 000122      MOV    (SP)+,DPRV4+2 ; RESTORE DPRV4.
177332 012637 000120      MOV    (SP)+,DPRV4
177336 011500      MOV    ($EF),R0     ; ERROR BITS => R0<11:00>...
177340 001402      BEQ    30010$        ;...AND SKIP IF NONE.
177342 052700 130000      BIS    #<13*BIT12>,R0 ; ELSE, ADD TEST NUM => R0<15:12>...
177346 000207      30010$: RETURN      ;...AND RETURN (NZ).

172
173 177350 012704 000004      IPV4:  MOV    #4.,R4        ; DPRV4 INTERRUPT RECEIVED.
174 177354 000405      BR     IPVXIT
175 177356 012704 000010      IPV8:  MOV    #8.,R4        ; DITTO DPRV8.
176 177362 000402      BR     IPVXIT
177 177364 012704 000014      IPV12: MOV   #12.,R4       ; DITTO DPRV12.
178 177370 042737 000007 177530      IPVXIT: BIC   #7,$CSR      ; *** TEMP, ENSURE REQ'S ARE CLEAR. ***
179 177376 000002      RTI

```

151

```

1 ;*****
2 ; SUBROUTINE TO ACCESS THE Q-BUS I/O PAGE USING THE DMA ENGINE.
3 ; THIS CODE IS PUSHED ONTO AND EXECUTED IN STACK SPACE.
4 ;
5 .DSABL AMA ; RELATIVE PIC.
6 ;
7 ; ON CALL, R0 = QIO ADDRESS, R1 = LOCAL ADDRESS, R2 = WORD COUNT.
8 ; ON RETURN, THEY CONTAIN UPDATED (TERMINAL) VALUES.
9 ; IF EOP TERMINATION, "V" BIT IS SET.
10 ;
11      174400          DM= SDMA           ; DMA CHIP BASE.
12 ;
13 177400 042767 000020 000170  RQIO: BIC   #BIT4,CM+2 ; READ -- CLEAR "FLIP" BIT.
14 177406 000403          BR    #QIOA
15 177410 052767 000020 000160  WQIO: BIS   #BIT4,CM+2 ; WRITE -- SET "FLIP" BIT.
16 177416 010067 000142  #QIOA: MOV   R0,QA+2 ; SET Q-BUS I/O ADDRESS.
17 177422 010167 000142          MOV   R1,LA+2 ; SET LOCAL ADDRESS.
18 177426 010267 000140          MOV   R2,WC ; SET WORD COUNT.
19 177432 012737 000115 174470  MOV   #115,@#DM+MMR ; MODE = VI, WAIT, CPINTLV, ENABLE.
20 177440 012737 000074 174454  MOV   #40134,@#DM+CMDR ; CLEAR INTERRUPT ENABLES.
21 177446 005037 174446          CLR   @#DM+CHAILH
22 177452 012700 000100          MOV   #QIOC--,-6,R0 ; GET (PIC) CHAIN START ADDRESS.
23 177456 060700          ADD   PC,PO
24 177460 000403          BR    CQIOA
25 177462 012700 000110  CQIO: MOV   #QIOC--,-6,R0 ; GET (PIC) CHAIN CONTINUE (DUMMY).
26 177466 060700          ADD   PC,RO
27 177470 010037 174442  CQIOA: MOV   R0,@#DM+CHAIL ; SET CHAIN ADDRESS.
28 177474 012737 000240 174454  MOV   #240,@#DM+CMDR ; CHAIN-LOAD (OR RELOAD 0) REGISTERS.
29 177502 012737 000102 174454  MOV   #102,@#DM+CMDR ; SET SOFT-REQ (START/CONTINUE DMA)...
30 177510 000240          NOP
31 177512 032737 020000 174456  1$:   BIT   #IP,@#DM+STAT1 ;...AND WAIT FOR DONE/ABORT (IP=1).
32 177520 001774          BEQ   1S
33 177522 012737 000054 174454  MOV   #40114,@#DM+CMDR ; OK, CLEAR IUS AND IP.
34 177530 013700 174412          MOV   @#DM+12,R0 ; GET FINAL Q ADDR (ARAL)...
35 177534 013701 174402          MOV   @#DM+2,R1 ;...LOCAL ADDR (ARBL)...
36 177540 013702 174462          MOV   @#DM+62,R2 ;...AND WORD COUNT (OPK).
37 177544 032737 000002 174456  BIT   #EOP,@#DM+STAT1 ; "EOP" TERMINATION ??
38 177552 001401          BEQ   2$ ; SKIP IF NOT...
39 177554 000262          SEV   ;...OTHERWISE, SET "V" IN PSW.
40 177556 000207          2$:   RETURN
41 ;
42 177560 001602          QIO: 1602           ; LOAD CARA, CARB, COPK, AND CR MODE.
43 177562 177400 000000  QA: 177400, 0 ; Q-BUS I/O ADDRESS (CARA)
44 177566 000000 000000  LA: 0, 0 ; LOCAL ADDRESS (CARB)
45 177572 000000          WC: 0 ; WORD COUNT (COPK)
46 177574 000010 001340  CM: 10, 1340 ; CHAN MODE (1360 IF "FLIP" = 1).
47 177600 000000          QIOC: 0 ; DUMMY RELOAD TO CLEAR "NAC" AND CONTINUE.
48 .ENABL AMA
49 ;*****

```

```
1 ; HI SEGMENT COPY OF ROM/RAM IN VECTOR SPACE ROUTINE.  
2 ;  
3 ; RETURN R2<00> = 1 IF LO BYTE NON-WRITABLE.  
4 ; R2<01> = 1 IF HI BYTE NON-WRITABLE.  
5 ;  
6 ;  
7 177602 005000 XRAMLO: CLR R0  
8 177604 005002 CLR R2  
9 177606 011046 MOV (R0),-(SP) ; GET TARGET LOCATION...  
10 177610 005116 COM (SP) ;...AND COMPLIMENT IT.  
11 177612 005110 COM (R0) ; COMPLIMENT TARGET LOCATION.  
12 177614 126066 000001 000001 CMPB 1(R0),1(SP) ; HI BYTE WRITABLE ??  
13 177622 001402 BEQ 1$ ; YES  
14 177624 052702 000002 BIS #BIT1,R2 ; NO  
15 177630 121026 1$: CMPB (R0),(SP)+ ; LO BYTE WRITABLE ??  
16 177632 001402 BEQ 2$ ; YES  
17 177634 052702 000001 BIS #BIT0,R2 ; NO  
18 177640 005110 2$: COM (R0) ; RESTORE TARGET LOCATION.  
19 177642 005702 TST R2 ; SET CONDITION FOR CALLER.  
20 177644 000207 RETURN  
21 ;  
22 ; HIGH SEGMENT COPY OF THE FLOATING I/O DATA TABLE.  
23 ;  
24 177646 015 012 XFLT10: .BYTE 15, 12 ; SYNC PAIR.  
25 177650 001 002 004 .BYTE 001, 002, 004, 010, 020, 040, 100, 200 ; FLOATING 1.  
26 177660 376 375 373 .BYTE 376, 375, 373, 367, 357, 337, 277, 177 ; FLOATING 0.  
27 177670 000000 .WORD 0 ; TERMINATOR.  
28  
29 .DSABL AMA
```

ABAUD	166656 G	BIT4 = 000020 G	CNTNU = 000062	DMA600	171160	IN300	173402	
AB100	166712	BIT5 = 000040 G	CQIOA	177462	DPRV12=	000134	IN400	173402
AB200	166722	BIT6 = 000100 G	CR	000015 G	DPRV4 =	000120	IP =	020000
AB300	166740	BIT7 = 000200 G	CRDS	166277 G	DPRV8 =	000124	IPVXIT	177370
AB400	166756	BIT8 = 000400 G	CRLF	166274 G	DS100	166346	IPV12	177364
AB500	166770	BIT9 = 001000 G	CSTMD	000006 G	DS200	166346	IPV4	177350
AB600	166770	BOOTFO 177774 G	CTCC	= 000040	ENDTAB	166655	IPV8	177356
ACC	161146	BOOTST 172322	CTCS	= 000024	EOP =	000002	IRX	163566
ACUM	173540 G	BRKENT 164674 G	CTMS	= 000070	EOPBIT=	000002	ISEXIT	163600
ADDRREG	165522 G	BSUPW 171170 G	CTRL	= 000000	ERRA =	100000 G	ITX	163544
ADSTKO	177700 G	BS100 171240	CTTC	= 000054	ERRFBG	100000 G	IXIT	163630
AD100	165530	BS200 171300	CTV	= 000010	ERROR	166264 G	KAT =	170002
AD200	165530	BS300 171326	C CLR	162200	ERRTB	172750	KCSDD	000022 G
AD220	165550	BS400 171346	C8255 =	177526 G	EXAMIN	165742 G	KCSDI	000006 G
AD240	165560	BS500 171352	DARFGB	000002 G	EXMDIS	165470	KCSED	000020 G
AD250	165564	BS550 171372	DATCOM	166404 G	EXTEST	172570 G	KCSEI	000004 G
AD253	165574	BS570 171404	DA100	166412	EX10	165766	KCSGS	000010 G
AD255	165606	BS600 171420	DA200	166412	EX1000	166224	KC\$MAX	000022 G
AD270	165614	BUFR1 = 150002	DA370	166426	EX1050	166242	KCSN0P	000000 G
AD280	165642	BUFR2 = 160002	DA400	166452	EX1200	166260	KC\$RD	000014 G
AD290	165644	BVTES = 170002	DA500	166460	EX1300	166262	KC\$RES	000002 G
AD300	165646	B1200 = 000022	DA600	166464	EX20	165776	KC\$SS	000012 G
AD400	165652	B2400 = 000032	DA700	166472	EX300	166020	KC\$WD	000016 G
AD500	165666	B300 = 000002	DC DAT	161614	EX400	166030	KC.COM	000036 G
AD600	165674	B4800 = 000042	DC L80	161672	EX500	166040	KC.EOM	004000 G
AD700	165720	B600 = 000012	DC RCW	161500	EX600	166040	KC.ICC	000040 G
AD800	165724	B9600 = 000052	DC X1T	161676	EX700	166056	KC.IDA	000100 G
AI	175010	CABIT = 010000	DC XMT	161406	EX750	166072	KC.IDR	000200 G
AI.B0 =	164377	CHA = 000003	DECTST	170072 G	EX760	166140	KC.LEW	003400 G
AO.BI =	165400	CHAINX 176100	DEPOS	166474 G	EX770	166146	KC.VEC	177400 G
APLPC0	177766 G	CHAIN1 175756	DE200	166526	EX800	166156	KESDMA	000014 G
APLPS0	177770 G	CHAIN2 176000	DE50	166512	EX900	166172	KE\$ILC	000006 G
APLRO0	177750 G	CHAIN4 176020	DINIT	173412 G	EX950	166222	KE\$ILL	000010 G
APLR10	177752 G	CHAIN5 176040	DISPRG	165450	EX970	166222	KE\$ILV	000012 G
APLR20	177754 G	CHAIN6 176060	DISPST	166346 G	PAR	164116 G	KESMAX	000014 G
APLR30	177756 G	CHA1H = 000046	DISPWD	166532 G	FARSTR	172370 G	KE\$NDA	000002 G
APLR40	177760 G	CHA1L = 000042	DI100	166550	FAS	171430 G	KE\$NDR	000004 G
APLR50	177762 G	CHA2H = 000044	DI200	166554	FCHFGB	000010 G	KE\$OK	000000 G
APLSP0	177764 G	CHA2L = 000040	DMACDA=	174454 G	FERRD	000017 G	KH.CMD	175001 G
ARDPGB	000004 G	CHB = 000004	DMAED	000002 C	FLT0	163700	KH.CSD	177531 G
A-SET	162130	CHEND 165742	DMAFD	171032 G	FLT1	163670	KH.IOC	177541 G
A4800	162126	CHKCHR 173614 G	DMAIMA=	174470 G	FLT10	163666	KH.SC2	175005 G
BBF10	000000 G	CHNOFO= 174442	DMAAV =	000164	FRERRB=	020000 G	KH.SC3	175007 G
BBF20	177776 G	CHNST0= 174446	DMAKIT	175720	FTLST	172736 G	KH.STA	175003 G
BEVNT =	000100	CHTAB 165734	DMA00A=	174442 G	GETIN	173552 G	KL.CMD	175000 G
BI	175024	CHX = 000004	DMA00A=	174442 G	GE100	173552	KL.CSA	177520 G
BITACKV=	000130	CH100 173652	DMA00A=	174442 G	GE200	173610	KL.CSB	177522 G
BIT0 =	000001 G	CH200 173670	DMA00A=	174442 G	GE300	173612	KL.CSC	177524 G
BIT1 =	000002 G	CH250 173702	DMA00A=	174446 G	GO	166772 G	KL.CSD	177530 G
BIT10 =	002000 G	CH300 173704	DMA00A=	174532 G	GP300	167176	KL.IOC	177540 G
BIT11 =	004000 G	CH400 173704	DMA1VA=	174530 G	HZ800	162124	KL.SC2	175004 G
BIT12 =	010000 G	CKSUM 161042	DMA100	171104	INDATA	166640	KL.SC3	175006 G
BIT13 =	020000 G	CLISP = 170006	DMA200	171114	INHIB =	170010	KL.STA	175002 G
BIT14 =	040000 G	CM	DMA300	171142	INIT	173302 G	KMSIDS	000360 G
BIT15 =	100000 G	CMDR = 000054	DMA400	171150	INITST	172320	KMSMAP	000016 G
BIT2 =	000004 G	CMREGA= 174454	DMA50	171046	IN100	173352	KM\$STF	000007 G
BIT3 =	000010 G	CNDFCB 000020 G	DMA500	171150	IN200	173356	KM\$SWS	000360 G

KXTNF - KXT11CA Native Firmware MACRO V05.00 00:51 Page 118-2
 Symbol table

KPSINI	000000 G	KXSDME	040000 G	KX\$T04	100010 G	ODTSTK	177746 G	PWRW = 000024
KPSNNC	000007 G	KXSDPE	000040 G	KX\$T05	100020 G	OPENDS	166301 G	PW050 173026
KPSPBS	000005 G	KXSDRT	000200 G	KX\$T06	100040 G	ORERRB=	040000 G	PW100 173050
KPSQD	000003 G	KXSEXO	100000 G	KX\$T07	100100 G	PACS =	000020	PW200 173112
KPSQTS	000002 G	KX\$FEF	000040 G	KX\$T10	100200 G	PADB =	000032	PW300 173124
KPSSTS	000001 G	KX\$FO1	000001 G	KX\$T11	100400 G	PAHS =	000102	PW500 173152
KPSWST	000004 G	KX\$FO2	000002 G	KX\$T12	101000 G	PAMS =	000100	PW600 173156
KR\$QRP	040000 G	KX\$FO3	000004 G	KX\$T13	102000 G	PARLBD	000016 G	PW700 173252
KR\$TMS	000001 G	KX\$FO4	000010 G	KX\$2EW	000002 G	PAV =	000004	QA 177562
KS.ALN	003400 G	KX\$F05	000020 G	K3K =	014000	PRCS =	000022	QADR = 170004
KS.DA	000200 G	KX\$F06	000040 G	LA	177566	PBDB =	000034	QAHI = 170002
KS.DBG	020000 G	KX\$F07	000100 G	LADR =	170006	PBHS =	000122	QALD = 170004
KS.DR	000040 G	KX\$F10	000200 G	LRTEST	163712 G	PBMS =	000120	QC0MIN 167460 G
KS.EOM	000100 G	KX\$F11	000400 G	LB100	163712	PBREB =	000002 G	QC100 167560
KS.ERC	000036 G	KX\$F12	001000 G	LEDTSD	000017 G	PBRF =	000070 G	QC120 167616
KS.ERR	100000 G	KX\$F13	002000 G	LF	000012 G	PRV =	000006	QC150 167626
KS.ON	040000 G	KXSG00	000010 G	LS0	000017 G	PCDB =	000036	QC200 167640
KW\$PAB	040000 G	KX\$INC	000004 G	LS1	000016 G	PCDD =	000014	QC300 167660
KW\$PB0	000001 G	KX\$I20	000010 G	LS10	000005 G	PCRLF	165510	QC350 167714
KW\$PB1	000002 G	KX\$I24	000020 G	LS11	000004 G	PERV	173000 G	QC380 167734
KW\$PB2	000004 G	KX\$I34	000040 G	LS12	000003 G	PEVNT =	000104	QC400 167736
KW\$PB3	000010 G	KX\$LD0	000001 G	LS13	000002 G	PIOAV =	000154	QC420 167776
KW\$PB4	000020 G	KX\$LD1	000002 G	LS14	000001 G	PIOAVA=	177004 G	QC440 170002
KW\$PB5	000040 G	KX\$LD2	000004 G	LS15	000000 G	PIOBV =	000160	QC445 170022
KW\$PB6	000100 G	KX\$LD3	000010 G	LS2	000015 G	PIOBVA=	177006 G	QC447 170024
KW\$PB7	000200 G	KX\$MP0	000002 G	LS3	000014 G	PIOB =	174404	QC450 170042
KW\$PC0	100000 G	KX\$MP1	000004 G	LS4	000013 G	PIOB1	174450	QC470 170050
KW\$PLD	001000 G	KX\$MP2	000010 G	LS5	000012 G	PIOB2	174532	QC50 167512
KW\$PLD	000400 G	KX\$N0P	000040 G	LS6	000011 G	PIOB3	174562	QC500 170052
KW\$POD	002000 G	KX\$NRF	010000 G	LS7	000010 G	PIOICA=	177000 G	QC600 170056
KW\$P1D	004000 G	KX\$NXM	100000 G	LS8	000007 G	PIOTVA=	177010 G	QC70 167524
KW\$P2D	010000 G	KX\$ODC	000010 G	LS9	000006 G	PIOXIT	174746	QIO 177560
KW\$P3D	020000 G	KX\$ODT	000001 G	MAINTB=	000004 G	PIT	174152	QIOC 177600
KW.CMD	175000 G	KX\$OHL	000020 G	MCC =	000002	PIT1	174222	QIRV = 000144
KW.CSA	177520 G	KX\$OMO	000001 G	MEM1	160304	PRIBD	000001 G	QIRVI = 000150
KW.CSB	177522 G	KX\$ORO	000002 G	MEM2	160312	PROC	167010 G	QIXIT 176372
KW.CSC	177524 G	KX\$PRO	000020 G	MEM3	160320	PROMPT	166266 G	QODTM 171432 G
KW.CSD	177530 G	KX\$PWF	001000 G	MEM3A	160324	PRTYV =	000114	QODTMD 000005 G
KW.DAT	000004 G	KX\$QIE	020000 G	MIC =	000000	PRO =	000000	QSRET 171444 G
KW.DCO	000000 G	KX\$QOF	002000 G	MMR =	000070	PR1 =	000040	QS100 171476
KW.DCO	175010 G	KX\$QRE	002000 G	MMRFCA=	174470	PR2 =	000100	QS1000 171710
KW.DC1	175020 G	KX\$QRF	004000 G	MODE =	170004	PR3 =	000140	QS1100 171726
KW.DST	000002 G	KX\$RTE	000100 G	NACBIT=	004000	PR4 =	000200	QS200 171476
KW.IOC	177540 G	KX\$R20	000001 G	NECV =	000070	PR5 =	000240	QS300 171536
KW.SC2	175004 G	KX\$R24	000002 G	NECXIT	163470	PR6 =	000300	QS400 171560
KW.SC3	175006 G	KX\$R34	000004 G	NEC.AA	162262	PR7 =	000340	QS500 171612
KW.STA	175002 G	KX\$SEF	000010 G	NEC.AS	162242	PUTEST	172324 G	QS700 171632
KX\$BAT	002000 G	KX\$SHC	000020 G	NEC.DA	162210	PUTSDS	000007 G	QS800 171656
KX\$BDF	020000 G	KX\$SSA	000004 G	NEC.DM	162624	PU100	172440	QS900 171702
KX\$BHE	000400 G	KX\$SOF	000100 G	NEC.EX	163640	PU1010	172566	RAMLO 160634
KX\$BHF	001000 G	KX\$SXT	004000 G	NEC.I	163522	PU20	172360	RAMLOA 160636
KX\$BQI	010000 G	KX\$TIS	000020 G	NEC.MD	163144	PU30	172372	RBOT1 000000 G
KX\$CME	100000 G	KX\$TRC	000001 G	NEC.SV	162250	PU400	172526	RBOT2 040000 G
KX\$CTE	000200 G	KX\$TTC	000010 G	NFERRD	000010 G	PU50	172434	RBOT3 100000 G
KX\$DEN	000100 G	KX\$T01	100001 G	NFFWD0	177772 G	PU500	172534	RBUFI1A= 177562 G
KX\$DEO	000004 G	KX\$T02	100002 G	NNCED	000000 G	PU600	172536	RCSR1A= 177560 G
KX\$DMC	000002 G	KX\$T03	100004 G	ODTSTG	177702 G	PWRUP	173012 G	RCVACB= 004000 G

RCVBKB=	004000 G	SN200	172262	S2ALBD	000014 G	TST6	161344	T7	- 160034 G
RCVDNB=	000200 G	SODTM	164626 C	S2BLBD	000015 G	TST7	161730	T7AETB	171002
RCVIEB=	000100 G	SODTMD	000004 G	TC	= 000001	TU100	171740	T7BFTB	171006
RD	= 000000	SO100	164746	TCLK	161164	TU1000	172150	T7ERR	170664 G
RDB	= 177776	SO1000	165104	TCPU	161150	TU120	171774	T7ERR1	170676
REGDIS	166303 G	SO1300	165142	TCXIT	161322	TU125	171776	T7ERR2	170704
REGFGB	000200 G	SO1400	165170	TDC	161344	TU130	172012	UROMOD	000011 G
REINIT	173016 G	SO1450	165174	TDMA	175106	TU150	172042	URUMID	000012 G
RESTRT	164000 G	SO1470	165174	TDMA1	175174	TU200	172060	VERS	173774 G
RE1000	164314	SO1500	165240	TDMA2	175276	TU300	172064	WAITD	000003 G
RE1010	164352	SO1550	165254	TDMA3	175346	TU400	172066	WAITST	167402 G
RE1020	164352	SO1552	165304	TDPR	176432	TU500	172102	WA100	167402
RE1040	164352	SO1555	165312	TDPR2	176520	TU58BT	171734 G	WA200	167446
RE1045	164372	SO1557	165322	TE100	172630	TU600	172110	WA300	167452
RF1050	164412	SO1560	165330	TE200	172642	TU800	172144	WC	177572
RE1060	164416	SO1565	165356	TE250	172712	TU900	172144	WCTBL	175742
RE1070	164416	SO1567	165364	TE300	172726	T0	= 160000 G	WQIO	177410
RE1072	164442	SO1569	165376	TE400	172740	T1	= 160004 G	WQIOA	177416
RE1075	164450	SO1570	165402	TIOR	160040	TIERK	170506 G	WRT	= 000010
RE1080	164450	SO1600	165404	TIPV1	176676	T1ERR1	170532	XBRKB	= 000001 G
RE200	164074	SO1700	165410	TIPV2	177100	T1ERR2	170540	XBUF1A	= 177566 G
RE250	164104	SO1800	165412	TIPXIT	177260	T10	= 160000 G	XCSR1A	= 177564 G
RE400	164106	SO1850	165412	TMP1	= 170002	T10ERR	170720 G	XCVR	= 000140
RE450	164120	SO1900	165440	TMP2	= 170004	T10ETB	171012	XDB	= 000002
RE500	164142	SO200	164746	TMP3	= 170006	T11	= 160004 G	XFLT10	177646
RE600	164226	SO300	164776	TMP4	= 170010	T11ERR	170726 G	XIEB	= 000100 G
RE700	164260	SO400	165006	TMP5	= 170012	T11ETB	171016	XOFF	= 000023
RE800	164270	SO700	165054	TMP6	= 170014	T12	= 160010 G	XOPFFGB	= 000040 G
RE900	164274	SO800	165060	TMP7	= 170016	T12FRR	170734 G	XON	= 000021
RNGFGB	000002 G	SO900	165062	TMP8	= 170020	T12ETB	171022	XRAMLO	177602
RQIO	177400	SPACE	166272 G	TNEC	161730	T13	= 160014 G	XRDVB	= 000200 G
RSTV	173004 G	SPOFF	000014 G	TNNCM1	170754	T13ERR	170742 G	SCSR	= 177530
RTOP1	077776 G	SPR5SU	173706 G	TNNCOM	170746	T13ETB	171026	SCSRA	= 177520
RTOP2	137776 G	SP1	077762 G	TNNINX	170762	T14	= 160020 G	SCSRB	= 177522
RTOP3	157776 G	SP2	137762 G	TNNXIT	170764	T15	= 160024 G	SCSRC	= 177524
SAVERG	164452 G	SP200	173724	TP10	174040	T16	= 160030 G	SCSRCO	= 177526
SD100	172300	SP3	157762 G	TOIR	176102	T17	= 160034 G	SDMA	= 174400
SEEND	172300	SP300	173752	TRAM	160272	T2	= 160010 G	SDPR	= 175000
SETIE	= 170010	SP400	173756	TRAPX	167366 G	T2ERR	170546 G	SEF	= \$000005
SETLDJ	167202 G	SR100	164474	TRAP10	167314 G	T2ERR1	170562	SEF1	= 177776
SETLED	167200 G	SR200	164526	TRAP24	167356 G	T3	= 160014 G	SEN	= 000014
SE100	167222	SR300	164536	TRAP4	167252 G	T3ERR	170576 G	SIPV	= 175000
SE150	167242	SR400	164622	TPOM	160700	T3ERR1	170612	SI8254	= 175720
SE200	167242	SSRET	164654 G	TR1010	167344	T3ERR2	170626	SLEDS	= 177524
SE300	167250	STACHN=	170012	TR410	167302	T4	= 160020 G	SPIO	= 177000
SKTA	160546	STAT	= 177774	TST1	160040	T4ERR	170642 G	QQDPR1	= 160000
SL1RV	= 000060	STATS	= 170006	TST10	174040	T4ETB	170766	QQDPR2	= 175400
SL1XV	= 000064	STAT1	= 000056	TST11	175106	T5	= 160024 G	SQIR	= 177532
SL2CAA	= 175704 G	STAT2	= 000054	TST12	176102	T5ERR	170650 G	SSL1	= 177560
SL2CBA	= 175714 G	STSENL	= 000010	TST13	176432	T5ETB	170772	SSL2	= 175700
SL2SAA	= 175700 G	STSTA8	170354	TST2	160272	T6	= 160030 G	STM	= 000013
SL2SBA	= 175710 G	S.SET	162150	TST3	160700	T6ERR	170656 G	STOP	= \$000005
SND8RK	172234	SOREGA	= 174456	TST4	161150	T6ETB	170776	SUID	= 177524
SN100	172240	SILBD	000013 G	TST5	161164				

* ABS. 177672 000 (RW,I,CBL,ABS,OVR)
 000000 001 (RW,I,LCL,REL,CON)