

```
;      A L T O C O N S T S 2 3 . M U
; Copyright Xerox Corporation 1979

; Symbol and constant definitons for the standard Alto microcode.
; These definitions are for:
;   AltoCode23, AltoCode24, AltoIICode2, and AltoIICode3
; By convention, people writing microcode should 'include' this file
;   in front of their microcode using the following MU construct:
;   #AltoConsts23.mu;
; This entire file is full of magic.  If you modify it in any way
;   you run the risk of being incompatible with the Alto world,
;   not to mention having your Alto stop working.
;
; Revision History:
; September 20, 1977  8:33 PM by Boggs
;   Created from old AltoConsts23.mu
; September 23, 1977 12:17 PM by Taft
; October 11, 1977  2:07 PM by Boggs
;   Added XMAR definition
; May 21, 1979  6:42 PM by Taft
;   Added SRB← and ESRB←
```

## ;Symbol definitions

## ;Bus Sources

```

;BS 0 ← RRegister
;BS 1 RRegister← (zeroes the bus)
;BS 2 is undefined and therefore makes the bus all ones
;BS 3 and 4 are task specific. For the 'Ram related' tasks they are:
;   BS 3: ← SRegister
;   BS 4: SRegister← (clobbers the bus with undefined value)
;BS 5 is main memory data (see definition for MD, below)
$MOUSE      $L000000,014006,000100; BS = 6
$DISP      $L000000,014007,000120; BS = 7

```

## ;Standard F1s

```

$XMAR      $L072000,000000,144000; F1 = 1 and F2 = 6 (Extended MAR)
$MAR       $L020001,000000,144000; F1 = 1
$TASK      $L016002,000000,000000; F1 = 2
$BLOCK     $L016003,000000,000000; F1 = 3
$LLSH1    $L000000,022004,000200; F1 = 4
$LRSH1    $L000000,022005,000200; F1 = 5
$LLCY8    $L000000,022006,000200; F1 = 6

```

## ;Standard F2s

```

$BUS=0     $L024001,000000,000000; F2 = 1
$SH<0     $L024002,000000,000000; F2 = 2
$SH=0     $L024003,000000,000000; F2 = 3
$BUS      $L024004,000000,000000; F2 = 4
$ALUCY    $L024005,000000,000000; F2 = 5
$MD       $L026006,014005,124100; F2 = 6, BS = 5

```

## ;Emulator specific functions

```

$BUSODD    $L024010,000000,000000; F2 = 10
$LMRSH1    $L000000,062005,000200; F2 = 11 Magic Right Shift
$LMLSH1    $L000000,062004,000200; F2 = 11 Magic Left Shift
$DNS       $L030012,000000,060000; F2 = 12 Do Nova Shift
$ACDEST    $L030013,032013,060100; F2 = 13 Nova Destination AC
$IR        $L026014,000000,124000; F2 = 14 Instruction Register
$IDISP     $L024015,000000,000000; F2 = 15 IR Dispatch
$ACSOURCE  $L000000,032016,000100; F2 = 16 Nova Source AC

```

## ;RAM-related task-specific functions

```

$SWMODE    $L016010,000000,000000; F1 = 10 Switch Mode (emulator only)
$WRTRAM    $L016011,000000,000000; F1 = 11 Write Ram
$RDRAM     $L016012,000000,000000; F1 = 12 Read Ram
$RRM       $L020013,000000,124000; F1 = 13 Reset Mode Register (emulator only)
$SRB       $L020013,000000,124000; F1 = 13 Set Register Bank (non-emulator)
$ESRB      $L020015,000000,124000; F1 = 15 Set Register Bank (emulator only)

```

## ;Emulator specific functions decoded by the ETHERNET board

```

$RSNF      $L000000,070016,000100; F1 = 16 Read Serial (Host) Number
$STARTF    $L016017,000000,000000; F1 = 17 Start I/O

```

```

$M         $R40; The M Register
$L         $L040001,036001,144200; The L Register
$T         $L052001,054001,124040; ALUF = 1, The T Register

```

## ;ALU Functions. \* =&gt; loads T from ALU output

```

$ORT       $L000000,050002,000002; ALUF = 2 *
$ANDT      $L000000,050003,000002; ALUF = 3
$XORT      $L000000,050004,000002; ALUF = 4
$+1        $L000000,050005,000002; ALUF = 5 *
$-1        $L000000,050006,000002; ALUF = 6 *
$+T        $L000000,050007,000002; ALUF = 7
$-T        $L000000,050010,000002; ALUF = 10
$-T-1     $L000000,050011,000002; ALUF = 11
$+INCT     $L000000,050012,000002; ALUF = 12 * synonym for +T+1
$+T+1     $L000000,050012,000002; ALUF = 12 *
$+SKIP     $L000000,050013,000002; ALUF = 13
$.T        $L000000,050014,000002; ALUF = 14 *
$AND NOT T $L000000,050015,000002; ALUF = 15

```

; A request has been made for the following, but it is unlikely ever to be implemented.

```

;$ZEROALU  $L000000,050016,000040; ALUF = 16

```

;ALUF 17 is unassigned

; Handy fakes

\$SINK \$L044000,000000,124000; DF3 = 0 Bus source without dest  
\$NOP \$L042000,000000,000000; NDF3 = 0 every computer needs one

; Definitions for the Nova debugger and DEBAL

\$HALT \$L042001,000000,000000;  
\$BREAK \$L042003,000000,000000;  
\$WENB \$L042005,000000,000000;  
\$READY? \$L042006,000000,000000;  
\$NOVA \$L044002,046003,124100;  
\$END \$L034000,000000,000000;

## ;Constant definitions

```

$0          $L000000,012000,000100; Constant 0 is SUPER SPECIAL

$ALLONES4   $M4:177777;      Constant normally ANDeD with KSTAT
$ALLONES5   $M5:177777;      Constant normally ANDeD with MD
$M17        $M6:000017;      Constant normally ANDeD with MOUSE
$ALLONES7   $M7:177777;      Constant normally ANDeD with DISP
$M177770    $M7:177770;      Mask for DISP
$M7         $M7:000007;      Mask for DISP
$X17       $M7:000017;      Mask for DISP

$ONE        $1;              The constant 1
$2         $2;
$-2        $177776;          - Disk header word count
$3         $3;
$4         $4;
$5         $5;
$6         $6;
$7         $7;
$10        $10;
$-10       $177770;          - Disk label word count
$17        $17;
$20        $20;
$37        $37;
$ALLONES   $177777;          The REAL -1 (not a mask)
$40        $40;
$77        $77;
$100       $100;
$177       $177;
$200       $200;
$377       $377;
$177400    $177400;          - DISK DATA WORD COUNT
$-400      $177400;
$2000      $2000;
$PAGE1     $400;
$DASTART   $420;             MAIN MEMORY DISPLAY HEADER ADDRESS
$KBLKADR   $621;             MAIN MEMORY DISK BLOCK ADDRESS
$MOUSELOC  $424;             MAIN MEMORY MOUSE BLOCK ADDRESS
$CURLOC    $426;             MAIN MEMORY CURSOR BLOCK ADDRESS
$CLOCKLOC  $430;
$CON100    $100;
$SCADM     $7772;           CYLINDER AND DISK MASK
$SECTMSK   $170000;         SECTOR MASK
$SECT2CM   $40000;          CAUSES ILLEGAL SECTORS TO CARRY OUT
$-4        $177774;          CURRENTLY UNUSED
$177766    $177766;          CURRENTLY UNUSED
$177753    $177753;          CURRENTLY UNUSED
$TOTUWC    $44000;          NO DATA TRANSFER, USE WRITE CLOCK
$TOWTT     $66000;          NO DATA TRANSFER, DISABLE WORD TASK
$STUWC     $4000;           TRANSFER DATA USING WRITING CLOCK
$STRWFSS   $10000;          TRANSFER DATA USING NORMAL CLOCK, WAIT FOR SYNC
$177000    $177000;
$77777     $77777;
$77740     $77740;
$LOW14     $177774;
$77400     $77400;
$-67D     $177675;
$7400      $7400;
$7417      $7417;
$170360    $170360;
$60110     $60110;
$30000     $30000;
$70531     $70531;
$20411     $20411;
$65074     $65074;
$41023     $41023;
$122645    $122645;
$177034    $177034;
$37400     $37400;
$BIAS      $177700;         CURSOR Y BIAS
$WNLOC     $452;           WAKEUP WAITING IN PAGE 1
$PCLOC     $500;           PC VECTOR IN PAGE 1
$100000    $100000;
$177740    $177740;

```

\$COMERR1	\$277;	COMMAND ERROR MASK
\$-7	\$177771;	CURRENTLY UNUSED
\$177760	\$177760;	
\$-3	\$177775;	
\$4560	\$4560;	
\$56440	\$56440;	
\$34104	\$34104;	
\$64024	\$64024;	
\$176000	\$176000;	
\$177040	\$177040;	
\$177042	\$177042;	
\$203	\$203;	
\$360	\$360;	
\$177600	\$177600;	
\$174000	\$174000;	
\$160000	\$160000;	
\$140000	\$140000;	
\$777	\$777;	
\$1777	\$1777;	
\$3777	\$3777;	
\$7777	\$7777;	
\$17777	\$17777;	
\$37777	\$37777;	
\$1000	\$1000;	
\$20000	\$20000;	
\$40000	\$40000;	
\$-15D	\$177761;	
\$TRAPDISP	\$526;	
\$TRAPPC	\$527;	
\$TRAPCON	\$470;	
\$JSRC	\$6000;	JSR@ 0
\$MASKTAB	\$460;	Mask Table Starting address for convert
\$SSH3CONST	\$14023;	DESTINATION = 3, SKIP IF NONZERO CARRY,
;		BASE CARRY = 0
\$600	\$600;	Ethernet addresses
\$601	\$601;	
\$602	\$602;	
\$603	\$603;	
\$604	\$604;	
\$605	\$605;	
\$606	\$606;	
\$607	\$607;	
\$610	\$610;	
\$612	\$612;	
\$ITQUAN	\$422;	
\$ITIBIT	\$423;	
\$402	\$402;	where label block is stored on disk boot
\$M177760	\$M7:177760;	MASK FOR DISP. FOR I/O INSTRUCTIONS
\$JSRCX	\$4000;	JSR 0
\$KBLKADR2	\$523;	
\$KBLKADR3	\$524;	
\$MFRRDL	\$177757;	DISK HEADER READ DELAY IS 21 WORDS
\$MFR0BL	\$177744;	DISK HEADER PREAMBLE IS 34 WORDS
\$MIRRD1	\$177774;	DISK INTERRECORD READ DELAY IS 4 WORDS
\$MIROBL	\$177775;	DISK INTERRECORD PREAMBLE IS 3 WORDS
\$MRPAL	\$177775;	DISK READ POSTAMBLE LENGTH IS 3 WORDS
\$MWPAL	\$177773;	DISK WRITE POSTAMBLE LENGTH IS 5 WORDS
\$BDAD	\$12;	ON BOOT, DISK ADDRESS GOES IN LOC 12
\$REFMSK	\$77740;	MRT Refresh mask
\$X37	\$M7:37;	NOPAR MASK
\$M177740	\$M7:177740;	DITTO
\$EIALOC	\$177701;	LOCATION OF EIA INPUT HARDWARE
\$7000	\$7000;	mapbase
\$176	\$176;	mapmask
\$177576	\$177576;	mapmask3
\$30	\$30;	reprobin
\$15	\$15;	wrt-1
\$1770	\$1770;	ciad
\$101771	\$101771;	cilow
\$175777	\$175777;	for resetting fbn.
\$11	\$11;	just to have small integers

\$13	\$13;	
\$14	\$14;	
\$16	\$16;	for 2CODE
\$60	\$60;	low R to high R bus source
\$776	\$776;	
\$177577	\$177577;	-129
\$100777	\$100777;	
\$177677	\$177677;	
\$177714	\$177714;	(-2fvar+14)
\$2527	\$2527;	
\$101	\$101;	
\$630	\$630;	
\$631	\$631;	
\$642	\$642;	
\$1gm1	\$M7:1;	
\$1gm3	\$M7:3;	
\$1gm10	\$M7:10;	
\$1gm14	\$M7:14;	
\$1gm20	\$M7:20;	
\$1gm40	\$M7:40;	
\$1gm100	\$M7:100;	
\$1gm200	\$M7:200;	
\$disp.300	\$M7:300;	
\$-616	\$177162;	
\$-650	\$177130;	
\$22	\$22;	
\$24	\$24;	
\$-20	\$177760;	
\$335	\$335;	endcode for getframe
\$1377	\$1377;	smallnzero
\$401	\$401;	
\$2001	\$2001;	
\$21	\$21;	just to have them
\$23	\$23;	
\$25	\$25;	
\$26	\$26;	
\$27	\$27;	
\$31	\$31;	
\$1675	\$1675;	
\$736	\$736;	
\$-660	\$177120;	
\$300	\$300;	
\$disp.377	\$M7:377;	
\$6001	\$6001;	f.e. flg, quick flg, use count
\$disp.3	\$M7:3;	

; Constants for subroutine returns using IR.  
; See 9.2.1 of the hardware manual for details.

\$sr1	\$60110;	
\$sr0	\$70531;	
\$sr2	\$61000;	
\$sr3	\$61400;	
\$sr4	\$62000;	
\$sr5	\$62400;	
\$sr6	\$67000;	value of 16b mapped to 6 by disp prom
\$sr7	\$63400;	
\$sr10	\$64024;	
\$sr11	\$64400;	
\$sr12	\$65074;	
; Are you wondering why sr13 is missing? So is everyone else.		
\$sr14	\$66000;	
\$sr15	\$66400;	
\$sr16	\$63000;	value of 6 mapped to 16b by disp prom
\$sr17	\$77400;	
\$sr20	\$65400;	
\$sr21	\$65401;	
\$sr22	\$65402;	
\$sr23	\$65403;	
\$sr24	\$65404;	
\$sr25	\$65405;	
\$sr26	\$65406;	
\$sr27	\$65407;	

```
$sr30      $65410;
$sr31      $65411;
$sr32      $65412;
$sr33      $65413;
$sr34      $65414;
$sr35      $65415;
$sr36      $65416;
$sr37      $65417;

$-13D      $177763;

$ERRADDR   $177024;      AltoII MEAR (Memory Error Address Reg)
$ERRSTAT   $177025;      AltoII MESR (Memory Error Status Reg)
$ERRCTRL   $177026;      AltoII MECR (Memory Error Control Reg)
$REFZERO   $7774;

$2377      $2377;      Added for changed Ethernet microcode
$2777      $2777;
$3377      $3377;
$477       $477;      Added for BitBlt
$576       $576;      Added for Ethernet boot
$177175    $177175;
```

```
;Requests for the following new constants have been made:
;NOTE THAT THESE ARE NOT YET DEFINED
```

```
;$1gm2     $M7:2;
;$1gm4     $M7:4;
;$32       $32;
;$33       $33;
;$34       $34;
;$35       $35;
;$36       $36;
```