Chipsmall
深圳市津利帝科技有限公司
Geelead Technology Limited
Chipsmall
采购部

JTAG-Booster for

Analog Devices ADSP-21xxx

Forth-Systeme GmbH

P.O: Box 1103 • D-79200 Breisach, Germany
Kueferstrasse 8 • D-79206 Breisach, Germany
Tel. +49 (7667) 908-0 • Fax +49 (7667) 908-200
sales@fsforth.de • http://www.fsforth.de

FS FORTH-SYSTEME GmbH
Postfach 1103, D-79200 Breisach, Germany

Release of Document:	April 19, 2002
Author:	Dieter Fögele
Filename:	JTAG_ADSP-21xxxa.doc
Program Version:	4.xx

Table of Contents

1. General

The programs JTAG060.EXE, JTAG065.EXE and JTAG161.EXE use the JTAG port of the Analog Devices ADSP-21xxx DSP Microcomputer in conjunction with the small JTAG-Booster:

- to program data into flash memory
- to verify and read the contents of a flash memory
- to make a memory dump
- to access an I²C Device
- to test CPU signals

All functions are done without any piece of software running in the target. No firmware or BIOS must be written. Bootstrap software may be downloaded to initially unprogrammed memories.

The JTAG-BOOSTER' s software is highly optimized to the JTAG chain of a specific target CPU. To give support for all DSPs of the Analog Devices ADSP-21xxx family, there are three different programs on the distribution disk:

- JTAG060.EXE : Tool for Analog Devices ADSP-21060/21061/21062
- JTAG065.EXE : Tool for Analog Devices ADSP-21065
- JTAG161.EXE : Tool for Analog Devices ADSP-21161

Please contact us, if you need support for other members of the Analog Devices ADSP-21xxx family.

For latest documentation please refer to the file README.TXT on the distribution disk.

1.1. Ordering Information

The following related products are available

- 940 JTAG-Booster Analog Devices ADSP-21xxx, 5V,
  DOS/Win9x/WinNT,
  delivered with adapter type 227

- 953 JTAG-Booster Analog Devices ADSP-21xxx, 3.3V,
  DOS/Win9x/WinNT,
  delivered with adapter type 285

1.2. System Requirements

To successfully run this tool the following requirements must be met:

- MSDOS, WIN3.x, WIN9x, WinME, WinNT or Win2000
  (WinNT/Win2000 is supported with an additional tool, see chapter 5)

- Intel 80386 or higher

- 205 kByte of free DOS memory

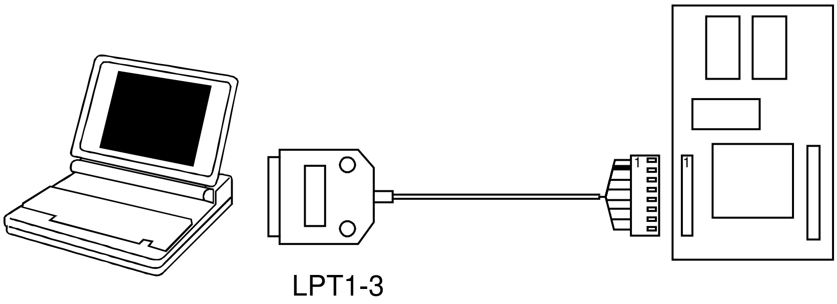
- Parallel Port

1.3. Contents of Distribution Disk

• JTAG060.EXE JTAG060.OVL	Tool for Analog Devices ADSP-21060/21061/21062
• JTAG060.INI	Template configuration file for Analog Devices ADSP-21060/21061/21062. See chapter 1.10 "Initialization file JTAGxxx.INI"
• JTAG065.EXE JTAG065.OVL	Tool for Analog Devices ADSP-21065
• JTAG065.INI	Template configuration file for Analog Devices ADSP-21065. See chapter 1.10 "Initialization file JTAGxxx.INI"
• JTAG161.EXE JTAG161.OVL	Tool for Analog Devices ADSP-21060/21061/21062
• JTAG161.INI	Template configuration file for Analog Devices ADSP-21161. See chapter 1.10 "Initialization file JTAGxxx.INI"
• HEX2BIN.EXE	Converter program to convert Intel HEX and Motorola S-Record files to binary. See chapter 4 "Converter Program HEX2BIN.EXE"
• WinNT.zip	Support for Windows NT and Windows 2000. See chapter 5 "Support for Windows NT and Windows 2000"
• JTAG_V4xx_FLASHES.pdf	List of all supported Flash devices
• README.txt	Release notes, new features, known problems

1.4. Connecting your PC to the target system

The JTAG-Booster can be plugged into standard parallel ports (LPT1-3) with a DB25-Connector.



The target end of the cable has the following reference:

1	2*	3	4	5	6	7	8
TCK	GND	TMS	TRST#	NC	TDI	TDO	+3.3V / +5V

*PIN 2 can be detected by the white thick cable.

To connect your design to the JTAG-BOOSTER you need a single row berg connector with a spacing of 2.54mm on your PCB. The names refer to the target: Pin 7 is the target's TDO pin and is connected to the JTAG-Booster's TDI pin.

There are two versions of the JTAG-Booster available: A 5V version (FS part number 227) and a 3.3V version (FS part number 285). **Don't apply 5V to the 3.3V version of the JTAG-Booster!**

Your target must be able to power the JTAG-Booster, it draws about 100mA.

Before you start the program, the JTAG-BOOSTER must be plugged to a parallel interface of your PC and to the 8 pin JTAG connector on the target.

The utility is started with the general command line format:

```
JTAGxxx /function [filename] [/option_1] ... [/option_n].
```

Note that the function must be the first argument followed (if needed) by the filename.

If you want to cancel execution of JTAGxxx, press CTRL-Break-Key.

On any error the program aborts with an MSDOS error level of one.

1.5. First Example with Analog Devices ADSP-21060/21061/21062

In the following simple example it is assumed that the JTAG-Booster is connected to LPT1 of your PC and target power is on.

Typing

```
        JTAG060  /P  MYAPP.BIN
```

at the DOS prompt results in the following output:

```
        JTAG060 --- JTAG utility for Analog Devices ADSP-21060/21061/21062
        Copyright © FS FORTH-SYSTEME GmbH, Breisach
        Version 4.xx of mm/dd/yyyy

(1)     Configuration loaded from file JTAGxxx.INI
(2)     Target: Analog Devices SHARC EZ-KIT
(3)     Using LPT at I/O-address 0378h
(4)     JTAG Adapter detected

(5)     1 Device detected in JTAG chain
          Device 0: Part without an ICODE register (1st Bit is 0)
(6)     Analog Devices ADSP-21060/21061/21062 is selected
(7)     Sum of instruction register bits  : 5
(8)     CPU position                      : 0
(9)     Instruction register offset       : 0
(10)    Length of boundary scan reg       : 363

        Looking for a known flash device. Please wait..
(11)    AMD 29F010 detected
(12)    Bus size is 8 Bit
(13)    Erasing Flash-EPROM Block #:0
        Programming File MYAPP.BIN
        65536 Bytes programmed
        Programming ok

        Erase Time          :          1.3 sec
        Programming Time    :        201.6 sec
```

(1) The initialization file JTAG060.INI was found in the current directory.

(2) The target identification line of the initialization file is printed here.

(3) The resulting I/O-address of the parallel port is printed here.

(4) A JTAG-Booster is found on the parallel port

(5) The JTAG chain is analyzed. There may be several parts in the JTAG chain. The chain is analyzed and all parts except the Analog Devices ADSP-21060/21061/21062 are switched to bypass mode.

(6) The Analog Devices ADSP-21xxx does not have an ICODE register.

(7) The length of all instruction registers in the JTAG chain are added.

(8) The position of the Analog Devices ADSP-21xxx in the JTAG chain is assumed to be zero, if not specified in the command line (see option /CPUPOS=).

(9) The position of the JTAG instruction register of the Analog Devices ADSP-21xxx is assumed to be zero, if not specified in the command line (see option /IROFFS=).

(10) The real length of the boundary scan register is displayed here and compared with the boundary scan register length of a Analog Devices ADSP-21060/21061/21062. This is very important, because the Analog Devices ADSP-21060/21061/21062 does not have an IDCODE register.

(11) One Flash-EPROM AMD 29F010 selected with BMS# was found.

(12) The resulting data bus size is printed here.

(13) In this example one block must be erased.

1.6. First Example with Analog Devices ADSP-21065

In the following simple example it is assumed that the JTAG-Booster is connected to LPT1 of your PC and target power is on.

Typing

```
JTAG065  /P  MYAPP.BIN
```

at the DOS prompt results in the following output:

```
        JTAG065 --- JTAG utility for Analog Devices ADSP-21065
        Copyright © FS FORTH-SYSTEME GmbH, Breisach
        Version 4.xx of mm/dd/yyyy

(1)     Configuration loaded from file JTAG065.INI
(2)     Target: Generic Target with ADSP-21065
(3)     Using LPT at I/O-address 0378h
(4)     JTAG Adapter detected

(5)     1 Device detected in JTAG chain
          Device 0: Part without an ICODE register (1st Bit is 0)
(6)     Analog Devices ADSP-21065 is selected
(7)     Sum of instruction register bits  : 5
(8)     CPU position                      : 0
(9)     Instruction register offset       : 0
(10)    Length of boundary scan reg       : 285

        Looking for a known flash device. Please wait..
(11)    AMD 29LV160B, 3.3V, Boot Block Bottom, byte mode detected
(12)    Bus size is 8 Bit
(13)    Erasing Flash-EPROM Block #:0  1  2  3
(14)    Unlock Bypass used
        Programming File MYAPP.BIN
        65536 Bytes programmed
        Programming ok

        Erase Time           :          0.9 sec
        Programming Time     :         81.5 sec
```

(1) The initialization file JTAG065.INI was found in the current directory.

(2) The target identification line of the initialization file is printed here.

(3) The resulting I/O-address of the parallel port is printed here.

(4) A JTAG-Booster is found on the parallel port

(5) The JTAG chain is analyzed. There may be several parts in the JTAG chain. The chain is analyzed and all parts except the Analog Devices ADSP-21065 are switched to bypass mode.

(6) The Analog Devices ADSP-21xxx does not have an ICODE register.

(7) The length of all instruction registers in the JTAG chain are added.

(8) The position of the Analog Devices ADSP-21xxx in the JTAG chain is assumed to be zero, if not specified in the command line (see option /CPUPOS=).

(9) The position of the JTAG instruction register of the Analog Devices ADSP-21xxx is assumed to be zero, if not specified in the command line (see option /IROFFS=).

(10) The real length of the boundary scan register is displayed here and compared with the boundary scan register length of a Analog Devices ADSP-21060/21061/21062. This is very important, because the Analog Devices ADSP-21060/21061/21062 does not have an IDCODE register.

(11) One Flash-EPROM AMD 29LV160B selected with BMS# was found.

(12) The resulting data bus size is printed here.

(13) In this example four blocks must be erased.

(14) If the used Flash-EPROM supports the unlock bypass feature, it is used to speed up programming performance.

1.7. First Example with Analog Devices ADSP-21161

In the following simple example it is assumed that the JTAG-Booster is connected to LPT1 of your PC and target power is on.

Typing

```
        JTAG161  /P  MYAPP.BIN
```

at the DOS prompt results in the following output:

```
        JTAG161--- JTAG utility for Analog Devices ADSP-21161
        Copyright © FS FORTH-SYSTEME GmbH, Breisach
        Version 4.xx of mm/dd/yyyy

(1)     Configuration loaded from file JTAG161.INI
(2)     Target: Generic Target with Analog Devices ADSP-21161
(3)     Using LPT at I/O-address 0378h
(4)     JTAG Adapter detected

(5)     1 Device detected in JTAG chain
          Device 0: Part without an ICODE register (1st Bit is 0)
(6)     Analog Devices ADSP-21161 is selected
(7)     Sum of instruction register bits  : 5
(8)     CPU position                      : 0
(9)     Instruction register offset       : 0
(10)    Length of boundary scan reg       : 481

        Looking for a known flash device. Please wait..
(11)    AMD 29LV160B, 3.3V, Boot Block Bottom, byte mode detected
(12)    Bus size is 8 Bit
(13)    Erasing Flash-EPROM Block #:0  1  2  3
(14)    Unlock Bypass used
        Programming File MYAPP.BIN
        65536 Bytes programmed
        Programming ok

        Erase Time          :          0.9 sec
        Programming Time    :         ??.0 sec
```

(1) The initialization file JTAG161.INI was found in the current directory.

(2) The target identification line of the initialization file is printed here.

(3) The resulting I/O-address of the parallel port is printed here.

(4) A JTAG-Booster is found on the parallel port

(5) The JTAG chain is analyzed. There may be several parts in the JTAG chain. The chain is analyzed and all parts except the Analog Devices ADSP-21161 are switched to bypass mode.

(6) The Analog Devices ADSP-21xxx does not have an ICODE register.

(7) The length of all instruction registers in the JTAG chain are added.

(8) The position of the Analog Devices ADSP-21xxx in the JTAG chain is assumed to be zero, if not specified in the command line (see option /CPUPOS=).

(9) The position of the JTAG instruction register of the Analog Devices ADSP-21xxx is assumed to be zero, if not specified in the command line (see option /IROFFS=).

(10) The real length of the boundary scan register is displayed here and compared with the boundary scan register length of a Analog Devices ADSP-21060/21061/21062. This is very important, because the Analog Devices ADSP-21060/21061/21062 does not have an IDCODE register.

(11) One Flash-EPROM AMD 29LV160B selected with BMS# was found.

(12) The resulting data bus size is printed here.

(13) In this example four blocks must be erased.

(14) If the used Flash-EPROM supports the unlock bypass feature, it is used to speed up programming performance.

1.8. Trouble Shooting

Avoid long distances between your Host-PC and the target. If you are using standard parallel extension cable, the JTAG-BOOSTER may not work. Don't use Dongles between the parallel port and the JTAG-BOOSTER.

Switch off all special modes of your printer port (EPP, ECP, ...) in the BIOS setup. Only standard parallel port (SPP) mode is allowed.

On very fast PCs there could be verify errors. To avoid this, watch for the ´IO recovery time´-switch in the BIOS Setup which must be turned on. Otherwise try to slow down your PC by setting the turbo switch off.

If there are problems with autodetection of the flash devices use the /DEVICE= option. To speed up autodetection specify one of the options /8BIT /16BIT or /32BIT.

Don't use hardware protected flash memories.

The used chip selects must be defined as output and inactive in the initialization file (see chapter 1.10 “Initialization file JTAGxxx.INI”). Also the address bits must be defined as output.

Use the option /NOWRSETUP to speed up flash programming.

1.9. Error Messages

- **80386 or greater required**
  The JTAG-BOOSTER does not work on a 8088/8086 or a 80286 platform.

- **Adapter not connected or target power fail**
  The JTAG-Booster wasn't found. Please check connection to parallel port and connection to target. Check target power. Check your BIOS-Setup.

- **Can't open x:\yyy\zzz\JTAGxxx.OVL**
  The overlay file JTAGxxx.OVL must be in the same directory as JTAGxxx.EXE.

- **Configuration file XYZ not found.**
  The file specified with the option /INI= wasn't found.

- **Device offset out of range**
  The value specified with the option /OFFSET= is greater than the size of the detected flash device.

- **Disk full**
  Writing a output file was aborted as a result of missing disk space.

- **Do not specify option /NOCS with any other chip select**
  There is a conflict in the command line.

- **Do not specify option /BYTE-MODE. *Flash device* does not have a byte mode pin.**
  The flash device specified with the option /DEVICE= does not support switching between 16 (or 32) bit mode and 8 bit mode. In practice it does not have a pin with the name BYTE#

- **Error creating file:**
  The output file could not be opened. Please check free disk space or write protection.

- **Error: *Pin-Name* is an output only pin**
  The specified pin cannot be sampled. Check the command line. Check the initialization file.

- **Error: *Pin-Name* is an input only pin**
  The specified pin cannot be activated. Check the command line. Check the initialization file.

- **Error: *Pin-Name* may not be read back**
  The specified pin can be switched to tristate, but cannot be read back. Check the command line.

- **illegal function:**
  The first parameter of the command line must be a valid function. See chapter 2 “JTAGxxx Parameter Description” for a list of supported functions.

- **illegal number:**
  The specified number couldn’t be interpret as a valid number. Check the relevant number base.

- **illegal option:**
  See chapter 2 “JTAGxxx Parameter Description” for a list of supported options.

- **illegal Pin Type:**
  The name specified with the option /PIN= must be one of the list of chapter 1.10 ”Initialization file JTAGxxx.INI”

- **illegal Flash Type:**
  The name specified with the option /DEVICE= must be one of the list of chapter 1.11 ”Supported flash devices”.

- **Input file not found:**
  The specified file cannot be found

- **Input file is empty:**
  Files with zero length are not accepted

- **" " is undefined**
  Please check the syntax in your configuration file. (See chapter 1.10 “Initialization file JTAGxxx.INI”).

- **LPTx not installed**
  The LPT port specified with /LPTx cannot be found. Please check the LPT port or specify a installed LPT port. Check your BIOS setup.

- **missing filename**
  Most functions need a filename as second parameter.

- **missing option /I2CCLK=**
  Some functions need the option /I2CCLK= to be defined.

- **missing option /I2CDAT=**
  Some functions need the option /I2CDAT= or the options /I2CDATO= and /I2CDATI= to be defined.

- **missing option /LENGTH=**
  Some functions need the option /LENGTH= to be defined.

- **missing option /PIN=**
  Some functions need the option /PIN= to be defined.

- **More than 9 devices in the JTAG chain or TDI pin stuck at low level**
  The JTAG chain is limited to 9 parts. Check target power. Check the target's TDO pin.

- **No devices found in JTAG chain or TDI pin stuck at high level**
  A stream of 32 high bits was detected on the pin TDI. TDI may stuck at high level. Check the connection to your target. Check the target power. Check the target's TDO pin.

- **Option /CPUPOS= out of range**
  The number specified with the option /CPUPOS= must be less or equal to the number of parts minus 1.

- **Option /IROFFS= out of range**
  Please specify a smaller value

- **Part at specified position is not a Analog Devices ADSP-21xxx**
  The option /CPUPOS= points to a part not a Analog Devices ADSP-21xxx

- **Pins specified with /I2CCLK= and /I2CDAT= must have different control cells**
  The pin specified with the option /I2CDAT= must be able to be switched to high impedance while the pin specified with option /I2CCLK= is an active output. See chapter 1.10 “Initialization file JTAGxxx.INI”.

- **Pins specified with /I2CCLK= and /I2CDATI= must have different control cells**
  The pin specified with the option /I2CDATI= must be able to be switched to high impedance while the pin specified with option /I2CCLK= is an active output. See chapter 1.10 “Initialization file JTAGxxx.INI”.

- **Pins specified with /I2CDATO= and /I2CDATI= must have different control cells**
  The pin specified with the option /I2CDATI= must be able to be switched to high impedance while the pin specified with option /I2CDATO= is an active output. See chapter 1.10 “Initialization file JTAGxxx.INI”.

- **Specify only one of that options:**
  Some options are exclusive (i.e. /8BIT and /16BIT). Don't mix them.

- **Sum of instruction register bits to low. Should be at least 5 bits for a Analog Devices ADSP-21xxx**
  The sum of all instruction register bits in the JTAG chain does not fit to the Analog Devices ADSP-21xxx. Check the target connection. Check the target CPU type. Check the settings for /IROFFS= and /CPUPOS= , if there are several parts in the JTAG chain.

- **Target no longer connected**
  There is a cyclic check of the JTAG chain. Check target power. Check target connection.

- **There are unknown parts in the JTAG chain. Please use the option /IROFFS= to specify the instr. reg. offset of the CPU.**
  If there are unknown parts in the JTAG chain, the program isn’t able to determine the logical position of the CPU’s instruction register.

- **There is no Analog Devices ADSP-21xxx in the JTAG chain**
  No Analog Devices ADSP-21xxx was found in the JTAG chain. Check the target power. Try with option /DRIVER=4 again.

- **Value of option /FILE-OFFSET out of range**
  The value of the option /FILE-OFFSET= points behind end of file.

- **wrong driver #**
  The value specified with the option /DRIVER= is out of range.

- **wrong Identifier (xxxx)**
  No valid identifier found. Check the specified chip select signal and the bus width. Try with the option /DEVICE= .

- **Wrong length of boundary scan register. Should be 363 for a Analog Devices ADSP-21060/21061/21062. (Should be 285 for a Analog Devices ADSP-21065/Should be 481 for a Analog Devices ADSP-21161.)**
  The length of the boundary scan register of the selected part (if there are more than one in the chain) does not fit to the Analog Devices ADSP-21xxx. Check the target connection. Check the target CPU type. Check the settings for /IROFFS= and /CPUPOS= , if there are several parts in the JTAG chain.

1.10. Initialization file JTAGxxx.INI

This file is used to define the default direction and level of all CPU signals. This file **must be carefully adapted** to your design with the Analog Devices ADSP-21xxx. The Target-Entry is used to identify your design which is displayed with most commands.

When the program JTAGxxx.EXE is started it scans the current directory for an existing initialization file named JTAGxxx.INI. If no entry is found the default values are used. You may also specify the initialization file with the option /INI= . If the specified file isn't found, the program aborts with an error message.

The CPU pins can also be used with the functions /BLINK (chapter 2.9), /PIN? (chapter 2.10) and /SAMPLE (chapter 2.11) to test the signals on your design.

The sample file below represents the values which are used for default initialization when no initialization file could be found in the current directory and no initialization file is specified with the option /INI=.

Changes to the structure of the file could result in errors. Remarks can be added by using //.

Sample File JTAG060.INI:

```
// Description file for Analog Devices ADSP-21xxx
Target: Analog Devices SHARC EZ-KIT
// All chip select signals are set to output and inactive.
// All signals should be defined. Undefined signals are set to their defaults.
// Pin names are defined in upper case.
// Low active signals are signed with a trailing #.

// Group A:  All pins in this group must be set to the same direction
//           This pins are bidirectional
L5CLK        Inp      // Link Port Clock
L5DAT0       Inp      // Link Port Data
L5DAT1       Inp      //
L5DAT2       Inp      //
L5DAT3       Inp      //

// Group B:  All pins in this group must be set to the same direction
//           This pins are bidirectional
L4CLK        Inp      // Link Port Clock
L4DAT0       Inp      // Link Port Data
L4DAT1       Inp      //
L4DAT2       Inp      //
L4DAT3       Inp      //

// Group C:  All pins in this group must be set to the same direction
//           This pins are bidirectional
L3CLK        Inp      // Link Port Clock
L3DAT0       Inp      // Link Port Data
L3DAT1       Inp      //
L3DAT2       Inp      //
L3DAT3       Inp      //

// Group D:  All pins in this group must be set to the same direction
//           This pins are bidirectional
L2CLK        Inp      // Link Port Clock
L2DAT0       Inp      // Link Port Data
L2DAT1       Inp      //
L2DAT2       Inp      //
L2DAT3       Inp      //
```

```
// Group E: All pins in this group must be set to the same direction
//          This pins are bidirectional
L1CLK        Inp      // Link Port Clock
L1DAT0       Inp      // Link Port Data
L1DAT1       Inp      //
L1DAT2       Inp      //
L1DAT3       Inp      //

// Group F: All pins in this group must be set to the same direction
//          This pins are bidirectional
L0CLK        Inp      // Link Port Clock
L0DAT0       Inp      // Link Port Data
L0DAT1       Inp      //
L0DAT2       Inp      //
L0DAT3       Inp      //

// Group G: All pins in this group must be set to the same direction
//          This pins are bidirectional
//          This group is switched between output/active and
//          input/tristate during programming of Flash-EPROMs
DATA0        Inp      // External Bus Data
DATA1        Inp      //
DATA2        Inp      //
DATA3        Inp      //
DATA4        Inp      //
DATA5        Inp      //
DATA6        Inp      //
DATA7        Inp      //
DATA8        Inp      //
DATA9        Inp      //
DATA10       Inp      //
DATA11       Inp      //
DATA12       Inp      //
DATA13       Inp      //
DATA14       Inp      //
DATA15       Inp      //
DATA16       Inp      // = Boot-ROM D0
DATA17       Inp      // = Boot-ROM D1
DATA18       Inp      // = Boot-ROM D2
DATA19       Inp      // = Boot-ROM D3
DATA20       Inp      // = Boot-ROM D4
DATA21       Inp      // = Boot-ROM D5
DATA22       Inp      // = Boot-ROM D6
```

```
DATA23        Inp       // = Boot-ROM D7
DATA24        Inp       //
DATA25        Inp       //
DATA26        Inp       //
DATA27        Inp       //
DATA28        Inp       //
DATA29        Inp       //
DATA30        Inp       //
DATA31        Inp       //
DATA32        Inp       //
DATA33        Inp       //
DATA34        Inp       //
DATA35        Inp       //
DATA36        Inp       //
DATA37        Inp       //
DATA38        Inp       //
DATA39        Inp       //
DATA40        Inp       //
DATA41        Inp       //
DATA42        Inp       //
DATA43        Inp       //
DATA44        Inp       //
DATA45        Inp       //
DATA46        Inp       //
DATA47        Inp       //

// Group H: All pins in this group must be set to the same direction
//          DMAG1#, DMAG2# and ADRCLK are tristateable outputs but may
//          not be read back.
//          PAGE, RD# and WR# are bidirectional pins.
//          This group is switched to output/active during programming of
//          Flash-EPROMs.
PAGE          Out,Lo    // DRAM Page Boundary
DMAG1#        Out,Hi    // DMA Grant 1, DMA Channel 7
DMAG2#        Out,Hi    // DMA Grant 2, DMA Channel 8
WR#           Out,Hi    // Memory Write Strobe = WE# of Flash-EPROM
RD#           Out,Hi    // Memory Read Strobe = OE# of Flash-EPROM
ADRCLK        Out,Lo    // Clock Output Reference
```

```
// Group I:  All pins in this group must be set to the same direction
//           This pins are bidirectional
//           For Flash programming this pins must be set to output.
SW#             Out,Hi    // Synchronous Write Select
MS0#            Out,Hi    // Memory Select Line
MS1#            Out,Hi    // Memory Select Line
MS2#            Out,Hi    // Memory Select Line
MS3#            Out,Hi    // Memory Select Line
ADDR31          Out,Lo    //
ADDR30          Out,Lo    //
ADDR29          Out,Lo    //
ADDR28          Out,Lo    //
ADDR27          Out,Lo    //
ADDR26          Out,Lo    //
ADDR25          Out,Lo    //
ADDR24          Out,Lo    //
ADDR23          Out,Lo    //
ADDR22          Out,Lo    //
ADDR21          Out,Lo    //
ADDR20          Out,Lo    //
ADDR19          Out,Lo    //
ADDR18          Out,Lo    //
ADDR17          Out,Lo    //
ADDR16          Out,Lo    //
ADDR15          Out,Lo    //
ADDR14          Out,Lo    //
ADDR13          Out,Lo    //
ADDR12          Out,Lo    //
ADDR11          Out,Lo    //
ADDR10          Out,Lo    //
ADDR9           Out,Lo    //
ADDR8           Out,Lo    //
ADDR7           Out,Lo    //
ADDR6           Out,Lo    //
ADDR5           Out,Lo    //
ADDR4           Out,Lo    //
ADDR3           Out,Lo    //
ADDR2           Out,Lo    //
ADDR1           Out,Lo    //
ADDR0           Out,Lo    //
```