



Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!



Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

Email & Skype: info@chipsmall.com Web: www.chipsmall.com

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China



JTAG-Booster for Intel StrongARM SA11x0



P.O: Box 1103
Kueferstrasse 8
Tel. +49 (7667) 908-0
sales@fsforth.de

- D-79200 Breisach, Germany
- D-79206 Breisach, Germany
- Fax +49 (7667) 908-200
- <http://www.fsforth.de>

Copyright © 1995..2002:

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

Release of Document: May 15, 2002
Author: Dieter Fögele
Filename: JTAG_SA-11x0b.doc
Program Version: 4.xx

All rights reserved. No part of this document may be copied or reproduced in any form or by any means without the prior written consent of FS FORTH-SYSTEME GmbH.

Table of Contents

1. General	4
1.1. Ordering Information	5
1.2. System Requirements	5
1.3. Contents of Distribution Disk	6
1.4. Connecting your PC to the target system	7
1.5. First Example with Intel StrongARM SA-1100.....	9
1.6. First Example with Intel StrongARM SA-1110.....	11
1.7. Trouble Shooting	13
1.8. Error Messages	14
1.9. Initialization file JTAG11x0.INI.....	19
1.10. Supported flash devices	29
2. JTAG11x0 Parameter Description	30
2.1. Program a Flash Device	33
2.2. Read a Flash Device to file.....	37
2.3. Verify a Flash Device with file	39
2.4. Dump target memory.....	41
2.5. Program an I ² C-Device	43
2.6. Read an I ² C-Device to file.....	45
2.7. Verify an I ² C-Device with file.....	47
2.8. Dump an I ² C-Device	49
2.9. Toggle CPU pins	51
2.10. Polling CPU pins	52
2.11. Polling CPU pins while the CPU is running	53
2.12. Show status of all CPU pins while the CPU is running	54
3. Implementation Information	56
4. Converter Program HEX2BIN.EXE.....	57
5. Support for Windows NT and Windows 2000	59
5.1. Installation on a clean system.....	59
5.2. Installation with already installed a previous version of Kithara.....	59
5.3. De-Installation version 5.xx:.....	60

1. General

The programs JTAG1100.EXE and JTAG1110.EXE use the JTAG port of the Intel StrongARM SA-11x0 embedded microprocessor 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 microprocessors of the Intel StrongARM SA-11x0 family, there are two different programs on the distribution disk:

- JTAG1100.EXE : Tool for Intel StrongARM SA-1100
- JTAG1110.EXE : Tool for Intel StrongARM SA-1110

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

1.1. Ordering Information

The following related products are available

- 929 JTAG-Booster Intel StrongARM SA-11x0, 3.3V, DOS/Win9x/WinNT, delivered with adapter type 285
- 268 Adapter Keith&Koep, adapts the 8 pin connector of JTAG-Booster to 10 pin connector of Keith&Koep Evaluation Platform "ARNOLD II"

1.2. System Requirements

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

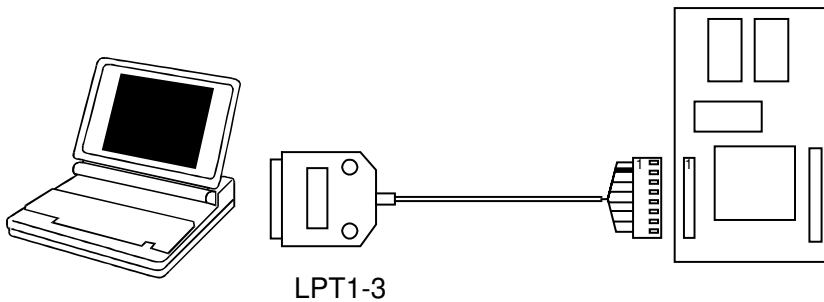
- MSDOS, WIN3.x, WIN9x, WinME, WinNT or Win2000 (WinNT/Win2000 is supported with an additional tool, see chapter 5)
- Intel 80386 or higher
- 205 kByte of free DOS memory
- Parallel Port

1.3. Contents of Distribution Disk

- JTAG1100.EXE Tool for Intel StrongARM SA-1100
 JTAG1100.OVL
- JTAG1100.INI Template configuration file for Intel StrongARM SA-1100. See chapter 1.9 "Initialization file JTAG11x0.INI"
- JTAG1110.EXE Tool for Intel StrongARM SA-1110
 JTAG1110.OVL
- JTAG1110.INI Template configuration file for Intel StrongARM SA-1110. See chapter 1.9 "Initialization file JTAG11x0.INI"
- HEX2BIN.EXE Converter program to convert Intel HEX and Motorola S-Record files to binary. See chapter 4 "Converter Program HEX2BIN.EXE"
- WinNT Support for Windows NT and Windows 2000. See chapter 5 "Support for Windows NT and Windows 2000"
- JTAG_V4xx_FLAS List of all supported Flash devices
 HES.pdf
- 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

*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.

The 3.3V version of the JTAG-Booster (FS part number 285) is delivered together with this package. Don't use the 5V version of the JTAG-Booster (FS number 227) with a 3.3V target. **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.

For the Keith&Koep Evaluation Platform "ARNOLD II" additional there is a 10 pin connector available to connect the JTAG-Booster (FS part number 268).

+3.3V	+3.3V	GND	GND	GND
2	4	6	8	10
1	3	5	7	9
TRST#	TDI	TDO	TMS	TCK

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:

JTAG11x0 /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 JTAG11x0, press CTRL-Break-Key.

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

1.5. First Example with Intel StrongARM SA-1100

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

```
JTAG1100 /P MYAPP.BIN
```

at the DOS prompt results in the following output:

```
JTAG1100 --- JTAG utility for Intel StrongARM SA-1100
Copyright © FS FORTH-SYSTEME GmbH, Breisach
Version 4.xx of mm/dd/yyyy

(1) Configuration loaded from file JTAG11x0.INI
(2) Target: Keith & Koep, ARNOLD II
(3) Using LPT at I/O-address 0378h
(4) JTAG Adapter detected

(5) 1 Device detected in JTAG chain
    Device 0: IDCODE=9108406B Intel StrongARM SA-1100, E-Step
(6) Sum of instruction register bits : 5
(7) CPU position                    : 0
(8) Instruction register offset     : 0
(9) Length of boundary scan reg    : 279

Looking for a known flash device. Please wait..
(10) Dual Intel 28F008 FlashFile Smart3/5 detected
(11) Bus size is 16 Bit
(12) Erasing Flash-EEPROM Block #:0
    Programming File MYAPP.BIN
    65536 Bytes programmed
    Programming ok

Erase Time           :          1.0 sec
Programming Time    :          30.3 sec
```

- (1) The initialization file JTAG1100.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 Intel StrongARM SA-1100 are switched to bypass mode.
- (6) The length of all instruction registers in the JTAG chain are added.
- (7) The position of the Intel StrongARM SA-11x0 in the JTAG chain is checked.
- (8) The position of the JTAG instruction register of the Intel StrongARM SA-11x0 is checked
- (9) The real length of the boundary scan register is displayed here and compared with the boundary scan register length of a Intel StrongARM SA-1100.
- (10) Two Flash-EPROMs Intel 28F008 selected with chip select CS0 are found.
- (11) The resulting data bus size is printed here.
- (12) In this example one block must be erased.

1.6. First Example with Intel StrongARM SA-1110

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

```
JTAG1110 /P MYAPP.BIN
```

at the DOS prompt results in the following output:

```
JTAG1110 --- JTAG utility for Intel StrongARM SA-1110
Copyright © FS FORTH-SYSTEME GmbH, Breisach
Version 4.xx of mm/dd/yyyy

(1) Configuration loaded from file JTAG1110.INI
(2) Target: Generic Target
(3) Using LPT at I/O-address 0378h
(4) JTAG Adapter detected

(5) 1 Device detected in JTAG chain
    Device 0: IDCODE=09261013 Intel StrongARM SA-1110, Revision 0
(6) Sum of instruction register bits : 5
(7) CPU position                    : 0
(8) Instruction register offset     : 0
(9) Length of boundary scan reg    : 292

Looking for a known flash device. Please wait..
(10) Dual Intel 28F008 FlashFile Smart3/5 detected
(11) Bus size is 16 Bit
(12) Erasing Flash-EPROM Block #:0
    Programming File MYAPP.BIN
    65536 Bytes programmed
    Programming ok

Erase Time      :      0.9 sec
Programming Time :     31.7 sec
```

- (1) The initialization file JTAG1110.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 Intel StrongARM SA-1110 are switched to bypass mode.
- (6) The length of all instruction registers in the JTAG chain are added.
- (7) The position of the Intel StrongARM SA-11x0 in the JTAG chain is checked.
- (8) The position of the JTAG instruction register of the Intel StrongARM SA-11x0 is checked
- (9) The real length of the boundary scan register is displayed here and compared with the boundary scan register length of a Intel StrongARM SA-1110.
- (10) Two Flash-EPROMs Intel 28F008 selected with chip select CS0 are found.
- (11) The resulting data bus size is printed here.
- (12) In this example one block must be erased.

1.7. 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 1.9 "Initialization file JTAG11x0.INI"). Also the address bits must be defined as output.

Use the option /NOWRSETUP to speed up flash programming.

1.8. 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\JTAG11x0.OVL**
The overlay file JTAG11x0.OVL must be in the same directory as JTAG11x0.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.
- **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#
- **Disk full**
Writing a output file was aborted as a result of missing disk space.
- **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 “JTAG11x0 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 “JTAG11x0 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 1.9 “Initialization file JTAG11x0.INI”
- **illegal Flash Type:**
The name specified with the option /DEVICE= must be one of the list of chapter 1.10 “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 1.9 “Initialization file JTAG11x0.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 Intel StrongARM SA-11x0**
The option /CPUPOS= points to a part not a Intel StrongARM SA-11x0

- **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 1.9 “Initialization file JTAG11x0.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 1.9 “Initialization file JTAG11x0.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 1.9 “Initialization file JTAG11x0.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 Intel StrongARM SA-11x0**
The sum of all instruction register bits in the JTAG chain does not fit to the Intel StrongARM SA-11x0. 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 Intel StrongARM SA-11x0 in the JTAG chain**
No Intel StrongARM SA-11x0 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 279 for a Intel StrongARM SA-1100. (Should be 292 for a Intel StrongARM SA-1110.)**
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 Intel StrongARM SA-11x0. 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.

1.9. Initialization file JTAG11x0.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 Intel StrongARM SA-11x0. The Target-Entry is used to identify your design which is displayed with most commands.

When the program JTAG11x0.EXE is started it scans the current directory for an existing initialization file named JTAG11x0.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 JTAG1100.INI:

```
// Description file for Intel StrongARM SA-1100
Target: Keith & Koep, ARNOLD II
// 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 group is switched between output/active and
//           input/tristate during programming of Flash-EPROMs
D0          Out,Lo //
D1          Out,Lo //
D2          Out,Lo //
D3          Out,Lo //
D4          Out,Lo //
D5          Out,Lo //
D6          Out,Lo //
D7          Out,Lo //
D8          Out,Lo //
D9          Out,Lo //
D10         Out,Lo //
D11         Out,Lo //
D12         Out,Lo //
D13         Out,Lo //
D14         Out,Lo //
D15         Out,Lo //
D16         Out,Lo //
D17         Out,Lo //
D18         Out,Lo //
D19         Out,Lo //
D20         Out,Lo //
D21         Out,Lo //
D22         Out,Lo //
D23         Out,Lo //
D24         Out,Lo //
D25         Out,Lo //
D26         Out,Lo //
D27         Out,Lo //
D28         Out,Lo //
D29         Out,Lo //
D30         Out,Lo //
```

```
D31          Out,Lo  //

// Group B: All pins in this group must be set to the same direction
//           For Flash programming, this group must be set to output
A0           Out,Lo  //
A1           Out,Lo  //
A2           Out,Lo  //
A3           Out,Lo  //
A4           Out,Lo  //
A5           Out,Lo  //
A6           Out,Lo  //
A7           Out,Lo  //
A8           Out,Lo  //
A9           Out,Lo  //
A10          Out,Lo  //
A11          Out,Lo  //
A12          Out,Lo  //
A13          Out,Lo  //
A14          Out,Lo  //
A15          Out,Lo  //
A16          Out,Lo  //
A17          Out,Lo  //
A18          Out,Lo  //
A19          Out,Lo  //
A20          Out,Lo  //
A21          Out,Lo  //
A22          Out,Lo  //
A23          Out,Lo  //
A24          Out,Lo  //
A25          Out,Lo  //
CS0#         Out,Hi  // = chip select 0
CS1#         Out,Hi  // = chip select 1
CS2#         Out,Hi  // = chip select 2
CS3#         Out,Hi  // = chip select 3
CAS0#        Out,Hi  //
CAS1#        Out,Hi  //
CAS2#        Out,Hi  //
CAS3#        Out,Hi  //
RAS0#        Out,Hi  //
RAS1#        Out,Hi  //
RAS2#        Out,Hi  //
RAS3#        Out,Hi  //
OE#          Out,Hi  // = output enable, is switched while accessing Flash
```

WE# Out,Hi // = write enable, is switched while accessing Flash

// The following pins are complete bidirectional pins.

// The direction of each pin can be set independent of the other pins.

// Each pin can be used as input.

L_FCLK	Inp
L_LCLK	Inp
LDD7	Inp
LDD6	Inp
LDD5	Inp
LDD4	Inp
LDD3	Inp
LDD2	Inp
LDD1	Inp
LDD0	Inp
L_PCLK	Inp
L_BIAS	Inp
GP0	Inp
GP1	Inp
GP2	Inp
GP3	Inp
GP4	Inp
GP5	Inp
GP6	Inp
GP7	Inp
GP8	Inp
GP9	Inp
GP10	Inp
GP11	Inp
GP12	Inp
GP13	Inp
GP14	Inp
GP15	Inp
GP16	Inp
GP17	Inp
GP18	Inp
GP19	Inp
GP20	Inp
GP21	Inp
GP22	Inp
GP23	Inp
GP24	Inp
GP25	Inp

```

GP26      Inp
GP27      Inp
TXD_C     Inp
RXD_C     Inp
SCLK_C    Inp
SFRM_C    Inp

```

// The following pins are output only pins.

// Setting to input (tristate) one of these pins results in an error.

```

POE#      Out,Hi // = PCMCIA output enable
PWE#      Out,Hi // = PCMCIA write enable
PIOR#     Out,Hi // = PCMCIA IO read
PIOW#     Out,Hi // = PCMCIA IO write
PSKTSEL   Out,Lo //
PREG#     Out,Hi //
PCE2#     Out,Hi //
PCE1#     Out,Hi //
RESET_OUT# Out,Hi // = Reset output, reset of Flashes connected

```

// The following pins are input only pins

// Setting to output one of these pins results in an error.

// Declaration of the direction of these pins is optional.

```

BATT_FAULT Inp //
VDD_FAULT  Inp //
IOIS16#    Inp //
PWAIT#     Inp //
RESET#     Inp //
ROMSEL     Inp //

```


Sample File JTAG1110.INI:

```
// Description file for Intel StrongARM SA-1110
Target: Generic Target
// 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 group is switched between output/active and
//           input/tristate during programming of Flash-EPROMs
D0          Out,Lo //
D1          Out,Lo //
D2          Out,Lo //
D3          Out,Lo //
D4          Out,Lo //
D5          Out,Lo //
D6          Out,Lo //
D7          Out,Lo //
D8          Out,Lo //
D9          Out,Lo //
D10         Out,Lo //
D11         Out,Lo //
D12         Out,Lo //
D13         Out,Lo //
D14         Out,Lo //
D15         Out,Lo //
D16         Out,Lo //
D17         Out,Lo //
D18         Out,Lo //
D19         Out,Lo //
D20         Out,Lo //
D21         Out,Lo //
D22         Out,Lo //
D23         Out,Lo //
D24         Out,Lo //
D25         Out,Lo //
D26         Out,Lo //
D27         Out,Lo //
D28         Out,Lo //
D29         Out,Lo //
D30         Out,Lo //
```

```
D31          Out,Lo  //
// Group B: All pins in this group must be set to the same direction
//           For Flash programming, this group must be set to output
A0           Out,Lo  //
A1           Out,Lo  //
A2           Out,Lo  //
A3           Out,Lo  //
A4           Out,Lo  //
A5           Out,Lo  //
A6           Out,Lo  //
A7           Out,Lo  //
A8           Out,Lo  //
A9           Out,Lo  //
A10          Out,Lo  //
A11          Out,Lo  //
A12          Out,Lo  //
A13          Out,Lo  //
A14          Out,Lo  //
A15          Out,Lo  //
A16          Out,Lo  //
A17          Out,Lo  //
A18          Out,Lo  //
A19          Out,Lo  //
A20          Out,Lo  //
A21          Out,Lo  //
A22          Out,Lo  //
A23          Out,Lo  //
A24          Out,Lo  //
A25          Out,Lo  //
CAS0#        Out,Hi  //
CAS1#        Out,Hi  //
CAS2#        Out,Hi  //
CAS3#        Out,Hi  //
RAS0#        Out,Hi  //
SDCAS#       Out,Hi  //
SDRAS#       Out,Hi  //
OE#          Out,Hi  // = output enable, is switched while programming
WE#          Out,Hi  // = write enable, is switched while programming
```