Digilent AVR Part Description File Specification

Revision: 2/6/06

www.digilentinc.com

215 E Main Suite D | Pullman, WA 99163
(509) 334 6306 Voice and Fax

Overview

The Digilent AVR Programmer™ uses part description files to program an AVR device. Part description files are XML documents that include device-specific information needed to program that AVR device as well as to display device-specific information in the Programmer's user interface. An example of a complete part description file is included in the appendix.

File Structure

All information in an XML file is contained in document elements delimited by tags. A tag has the form <TAGNAME>, where a tag name string is delimited by '<' and '>' with no spaces. A tagged section begins with a tag and ends with an end-tag. An end-tag has the form </TAGNAME>, which matches the beginning tag except that the tagname is preceded by a '/' character. The body of a tagged section may be made up of text or other tagged sections. All tagged sections must be fully nested.

The entire contents of the AVR part description file is contained within a tag called AVRPART.

```
<AVRPART>
    document body
</AVRPART>
```

The body of the AVRPART section is made up of five sections:

```
<FORMAT>
<ADMIN>
<MEMORY>
<FUSE>
<LOCKBIT>
```

These sections do not need to be in any particular order, but the above order is recommended.

FORMAT Section

This section is used to indicate that this is a Digilent format file. The body of this section is made up of the text "Digilent".

```
<FORMAT>Digilent</FORMAT>
```

ADMIN Section

This section is used to specify a data set name for the contents of the file, the device name for the specific AVR device that the file describes, and the signature bytes for the device. It contains the following three tags:

<name>
<device>
<signature>

NAME tag: This tag is used to specify a data set name for the file contents. This is currently not used by the AVR Programmer.

DEVICE tag: This tag is used to specify the name of the AVR device described by the contents of the file.

SIGNATURE tag: This tag is used to specify the signature bytes for the AVR device. The body of this section is made up of three hexadecimal bytes separated by spaces.

Example ADMIN section:

```
<ADMIN>
   <name>Digilent Cerebot</name>
   <device>ATmega64</device>
   <signature>0x1E 0x96 0x02</signature>
</ADMIN>
```

MEMORY Section

This section is used to specify the sizes of the memories in the device. This section is made up of the following tags:

<flash>
<flash_page>
<eeprom>

FLASH tag: This tag is used to specify the size of the program flash memory. The body of this tag contains a single decimal number giving the total size of the device program flash memory in bytes.

FLASH_PAGE tag: This tag is used to specify the size of the program flash page buffer. The body of this tag contains a single decimal number giving the flash page buffer size in 16 bit words. For AVR devices that aren't paged, the value should be 1.

EEPROM tag: This tag is used to specify the size of the data EEPROM memory. The body of this tag contains a single decimal number giving the total size of the data EEPROM memory in bytes.

Example MEMORY section:

```
<MEMORY>
   <flash>65536</flash>
   <flash_page>128</flash_page>
   <eeprom>2048</eeprom>
</MEMORY>
```

FUSE and LOCKBIT Sections

The FUSE section is used to describe the contents of the fuses list box in the Fuses tab of the AVR Programmer. The LOCKBIT section is used to describe the contents of the Lock Bits tab of the AVR programmer. The user interface elements that populate these list boxes are described in these sections. This section also specifies the number of fuse bytes and the default values for the fuse bytes.

The FUSE and LOCKBIT sections are made up of user interface element declarations, user interface element instantiations, and housekeeping values.

The user interface element declarations will declare one or more of the following interface elements:
<toggle>, <enum>, or <cond>

These are described fully in the user interface element declarations section of this document.

Following the user interface element declarations come the user interface element instantiations. These instantiations are made up of one or more <citm> tags. Each <citm> tag specifies an element to be added to the user interface list box and references a user interface element declared using one of the declarations described above.

Following the user interface element instantiations, the FUSE section contains a <count> tag and a <default> tag. These specify the number of fuse bytes in the device and the default value for the fuse bytes. The LOCKBIT section contains only a <default> tag.

COUNT tag: This tag specifies the number of fuse bytes in the device. The contents of this tag is a single decimal number in the range 1-3. For example:

```
<count>3</count>
```

DEFAULT tag: This tag specifies the default values for the fuse bytes or the lock bits. The contents of this tag will be 1 to 3 hexadecimal byte values separated by spaces. For example:

```
<default>0xFF 0xD9 0xE4</default>
```

The default tag for the fuses will specify as many values as the <count> tag indicated. The default value for the lock bits is a single byte value.