Chapter 4

BST DESIGN LANGUAGES

This chapter describes various software tools developed to support automation in Boundary-Scan testing. These tools support the features of Boundary-Scan applications in various phases of the product life cycle, where 'product' may be defined as IC, PCB or system. Table 4-1 intends to place the various tools in their right context, which may be useful when reading this chapter.

<table>
<thead>
<tr>
<th></th>
<th>Documentation/ Description</th>
<th>Design</th>
<th>Design Verification</th>
<th>Test Preparation</th>
</tr>
</thead>
<tbody>
<tr>
<td>IC</td>
<td>BSDL (VHDL)</td>
<td>Automatic BSC inserter (TIM / BSR compiler)</td>
<td>Conformance with Logic Analyzer</td>
<td>ATPG for inserted BS logic</td>
</tr>
<tr>
<td>PCB</td>
<td>EDIF</td>
<td>--</td>
<td>Disassembly</td>
<td>BS Test Pattern Gen.</td>
</tr>
<tr>
<td>System</td>
<td>--</td>
<td>--</td>
<td>Disassembly</td>
<td>BS Test Pattern Gen.</td>
</tr>
</tbody>
</table>

BSDL DESCRIPTION

The Boundary-Scan Description Language (BSDL) is formally presented as Supplement (B) to the IEEE Std 1149.1-1990. [18] and [19].

BSDL describes the testability features of Boundary-Scan devices which are compatible with the IEEE Std 1149.1. It is written within a subset of the VHSIC Hardware Description Language VHDL [20]. As such, BSDL is in itself not a general purpose hardware description language, but it can be used in conjunction with software tools for test generation, analysis and failure diagnosis. The parameters used in a BSDL description are orthogonal to the rules of the IEEE Std 1149.1. This means that elements of a design which are absolutely mandatory for the IEEE Std 1149.1 are not included in the language. This avoids ambiguous
descriptions and definitions. For example the BYPASS register is fully and without options described in the IEEE Std 1149.1 and hence it is not described in BSDL. Also, BSDL is not intended to describe (parts of) the on-chip system logic, but merely the properties of the boundary register with its terminal connections.

As stated above, BSDL comes as a subset of VHDL, in a case-insensitive free-form multi-line terminated form. The Backus-Naur Format (BNF) is used to describe the syntax. Comments are enclosed between a "--" and an EOL (end of line) character. For long strings a concatenation character "&" is used to break the strings in arbitrary but readable form. The concatenation character has no syntactical meaning and can be thought of as used in a lexicographical preprocessing step before the parsing process starts.

BSDL may be used in two environments: in a full or in a partial VHDL environment. In a full VHDL-based system the BSDL information is passed through the VHDL analyzer into a compiled design library, from where the boundary scan data are extracted by referencing the appropriate attributes. In the latter case only a limited set of VHDL syntax can be parsed. A fully integrated VHDL environment is supposed in the following descriptions.

BSDL comprises three main sections: the Entity, the Package and the Package Body. These "packages" remain constant along with the IEEE Std 1149.1. However, the user may add his own application specific "packages" to the standard formats.

The Entity Section

The Entity describes the Boundary-Scan parameters of a device’s I/O ports and attributes in terms of VHDL, with the specific IEEE Std 1149.1 related definitions coming from a pre-written VHDL standard Package and Package Body. The Entity in BSDL has the following structure.

```vhdl
entity My_Ic is -- an entity for my IC
  [generic parameter]
  [logical port description]
  [usage statement(s)]
  [package pin mapping]
  [scan port identification]
  [TAP description]
  [Boundary Register description]
end My_Ic; -- End description
```

This structure should be maintained with the order of elements as shown here. The elements will be addressed in the next subsections.