ASCII File Formats - Details

There are 3 ASCII file formats used for Cohesion Designer, described in this document:

- → Symbol Data
- → Schematic Data
- → Netlist Data

Symbol Data

This data file consists of all necessary information to re-construct a symbol in its binary format. It is therefore useful for ASCII backups and perl-based parsing of symbol data. This format is generated [from .sym files] by ‘asyout’, and read [back into .sym files] by ‘asyin’.

```
SymbolType %s
type ("", "COMPONENT", "GATE", "CELL", "BLOCK", "GRAPHIC", "PIN",
"MASTER", "RIPPER")
```

```
SYMATTR %s %s
attrName, value
```

```
SYMATTR %d %s
attrNumber, value
```

```
PINATTR %s %s
attrName, value
```

```
PINATTR %d %s
attrNumber, value
```

```
PIN %d %d  %s %d
x0, y0, justification ("NONE", "BOTTOM", "LEFT", "RIGHT", "TOP", "",
"VBOTTOM", "VLEFT", "VRIGHT", "VTOP"), nameOffset
```

```
WINDOW %d %d %d %s %d
number, x0, y0, justification ("Left", "Center", "Right", "VLeft",
"VCenter", "VRight"), fontSize
```

```
ARC %s %d %d %d %d %d %d %d %d %s
x0, y0, x1, y1, x2, y2, x3, y3, style ("", "Dash", "Dot", "DashDot",
"DashDotDot")
```

```
CIRCLE %s %d %d %d %d %s
x0, y0, x1, y1, style ("", "Dash", "Dot", "DashDot", "DashDotDot")
```

```
RECTANGLE %s %d %d %d %d %s
x0, y0, x1, y1, style ("", "Dash", "Dot", "DashDot", "DashDotDot")
```

```
LINE %s %d %d %d %d %s
x0, y0, x1, y1, style ("", "Dash", "Dot", "DashDot", "DashDotDot")
```

```
TEXT %d %d %s %d %s
x0, y0, justification ("Left", "Center", "Right", "VLeft",
"VCenter", "VRight"), fontSize, string
```

Schematic Data

This data file consists of all necessary information to re-construct a schematic cell in its binary format (without symbols). It is therefore useful for ASCII backups and perl-based parsing of schematic data. This format is generated [from .sch files] by ‘ascout’, and read [back into .sch files] by ‘ascin’.

Nets

`WIRE %d %d %d %d`
x0, y0, x1, y1

`FLAG %d %d %s`
x0, y0, name

`IOPIN %d %d %s`
x0, y0, type ("", "In", "Out", "BiDir")

`BUSTAP %d %d %d %d`
x1, y1, x0, y0 (1: bus trunk, 0: wire tip)

`ATTRFLAG %d %d`
x0, y0

`NETATTR %d %d %s %s`
x0, y0, AttrName (list attribute by name), value

`NETATTR %d %d %d %s`
x0, y0, AttrNumber (list attribute by number), value

Symbol Instances

`SYMATTR %s %s`
AttrName (list attribute by name), value

`SYMATTR %d %s`
AttrNumber (list attribute by number), value

`PINATTR %d %d %s %s`
x0, y0 (XY of pin), AttrName (list attribute by name), value

`PINATTR %d %d %d %s`
x0, y0 (XY of pin), AttrNumber (list attribute by number), value

`WINDOW %d %d %d %s %d` (overrides only, not for defaults)
WindowNumber, x0, y0, justification ("Left", "Center", "Right", "VLeft", "VCenter", "VRight"), fontSize

`SYMBOL %s %d %d %s` (symbol instance)
name, x0, y0, rotate/mirror ("R0", "R90", "R180", "R270", "M0", "M90", "M180", "M270")

Data Tables

`TABLE %d %d %d %d %d %d %d %d`
x0, y0, rows, columns, rowHeight, columnWidth, row0Height, column0Width

`TABLEATTR %d %s`
attrNumber, value

`TABLEDATA %d %d %s`
row, column, value

Graphics

`ARC %s %d %d %d %d %d %d %d %d %s`
x0, y0, x1, y1, x2, y2, x3, y3, style ("", "Dash", "Dot", "DashDot", "DashDotDot")

`CIRCLE %s %d %d %d %d %s`
`x0, y0, x1, y1, style ("", "Dash", "Dot", "DashDot", "DashDotDot")`

`RECTANGLE %s %d %d %d %d %s`
`x0, y0, x1, y1, style ("", "Dash", "Dot", "DashDot", "DashDotDot")`

`LINE %s %d %d %d %d %s`
`x0, y0, x1, y1, style ("", "Dash", "Dot", "DashDot", "DashDotDot")`

`TEXT %d %d %s %d %s`
`x0, y0, justification ("Left", "Center", "Right", "VLeft", "VCenter", "VRight"), fontSize, string`

Sheets

`SHEET %d %d %d`
`width, height, number`

Netlist Data

A netlist file contains all information related to connectivity, including attributes, but it does not contain “graphical” information. The netlist is written by ‘asciinet’ from hierarchy navigator, and ‘sasciinet’ as a stand-alone command-line process. The netlist is read by the schematic generator ‘scgen’.

There are two modes of the netlist: flattened and hierarchical. The difference between the two formats is only in the use of hierarchical constructs “cell”, “innet”, “outnet”, and “binet”. In a flattened netlist, only one “cell” is used, at the top of the netlist. In a hierarchical netlist, all constructs may be used.

Comments

`COMMENT <to line end>`

Cells

`CELL <cell name>`
Cell name is the name of the schematic. It does not refer to an alternate name such as the spice model, it is the root symbol name matching the .sym object that is the parent for this schematic, and also the name of the .sch object that is this schematic. It is used without any filename extension.

`INNET <name>`
`OUTNET <name>`
`BINET <name>`
Three types of I/O ports for a cell. These keywords identify both a port, as well as its net within the cell.

`NET <name>`
Statement for an internal net that does not leave the cell in an upward direction. This net may leave the cell in a downward direction, by connecting to a child instance.

`NETATTR <attribute name> <value>`
Assignment of an attribute to this net (both I/O and local nets). The attribute name must be made without spaces. The attribute value is any string to the line end, and may include spaces. In the case of derived attributes, the raw equation string is used in its unresolved state.

Instances

INSTANCE <cell name> <instance name>
Cell name is the name of the .sym object that is instantiated. It does not refer to an alternate name such as the spice model. Instance name is the unique handle to the object in schematic, similar to a spice instance name but not necessarily the same.

SYMATTR <attribute name> <value>
Assignment of an attribute to this instance. The attribute name must be made without spaces. The attribute value is any string to the line end, and may include spaces. In the case of derived attributes, the raw equation string is used in its unresolved state.

PIN <pin name> <connected net name>
Connection of an instance pin to a net. Also serves as the section header for a set of pin attributes. The keyword “<Unconnected>” may appear for explicit no-connects.

PINATTR <attribute name> <value>
Assignment of an attribute to this pin. The attribute name must be made without spaces. The attribute value is any string to the line end, and may include spaces. In the case of derived attributes, the raw equation string is used in its unresolved state.