dimensional orthogonal arrays), and omit more complex iterations such as hexagonal arrays. The more complex designs will simply have to call symbols explicitly, rather than using an iterative construct.

This list will doubtless grow as experience with CIF grows. Please help improve CIF by sending us comments and suggestions.

7.2 Ways to Generate CIF

CIF is an interchange format, usually generated by a computer-aided design system (Figure 7.1). There are thus as many ways to generate CIF as there are design systems. This section simply surveys some of the techniques.

7.2.1 Keyboard Interface

For small-scale projects, it is possible to generate a CIF file directly with an interactive text editor. The designer will generally make a detailed drawing of the layout, and read off coordinates from his drawing and enter the appropriate CIF commands one at a time. After the design is entered, it must be check-plotted in some way to verify that the file is in proper CIF format and correctly encodes the layout. In this way, a person can design a small experiment with minimal facilities.

Creating CIF files directly is much easier if the designer has a library of CIF text that defines useful symbols: pads, pad drivers, PLA cells, shift register cells, and so forth. The symbols must be accompanied by documentation that illustrates how to make connections to them, etc. Such a library is a tremendous aid to a beginning designer because it allows him to concentrate on his design ideas rather than on details of pad drivers and the like. The library is, in effect, a primitive design data base. Although CIF is not intended to be used for complex data bases, it can be used very effectively in this crude form.

Direct creation of CIF files can also be made more convenient and less error prone with a suitable preprocessor program. For example, an interactive input program might allow the designer to use relative coordinates, provide suitable default values, permit convenient iteration of cells, and check for obvious errors. It might prompt the user for the next entry and give feedback as to the current symbol definition or current layer. Ideally it would interact with the user in terms of symbolic symbol names and handle the generation of proper symbol numbers internally. (See section 2.2 for a further discussion of such a program.)

7.2.2 Programming Languages

If the designer is also a computer programmer, he may prefer to write a program that, when executed, writes a CIF file. This allows all the facilities offered in programming languages to be used in determining the geometry: arithmetic expressions, conditionals, procedures, etc. Often a cell is used with some minor modifications in many different places of the overall design. This cell could be defined by a procedure, with parameters that govern the modifications. The procedure in turn calls on "primitive" procedures that generate CIF geometric primitives directly. These primitive procedures might also generate a display or check-plot at the same time the CIF file is being constructed. (See [Newman & Sproull 1979] for a discussion of "display procedures." See [Locanthi 1978] for an example of a simple set of procedures for IC design.)

If a fully interactive programming environment is available, the user can change the program and re-execute it very quickly. Most BASIC, APL and LISP systems are interactive in this way, and do not incur the delay of compiling the program before it can be executed. If the execution of the program generates a display, the designer can very quickly modify the program to achieve the design he desires.

An important advantage to this approach is that the programs and procedures created by the designer can be saved for use in subsequent designs. Procedures corresponding to cell designs can be generalized and parameterized to apply in many situations. In effect, the collection of procedures becomes a very powerful design data base.

7.2.3 Interactive Graphical Layout Systems

A common design tool in use in the IC industry today is the interactive graphical layout system, for example the commercially available Calma or Applicon systems. A user views a display of all or part of his design, and requests changes by entering commands with a keyboard and graphical input device such as a tablet. The graphical editing commands are designed to make simple and complex changes naturally. The system should allow the user to insert and move geometrical shapes as easily as moving cardboard pieces on a floor plan. With the same ease the user should be able to change the shape of geometrical features. If the system automatically snaps geometrical shapes to a selectable grid and aligns all shapes parallel to the coordinate axes (or possibly lines at 45 degrees), it frees the designer from the tiring task of paying attention to exact coordinate values.

Present-day graphics systems are not without their shortcomings. Often it is cumbersome to define a collection of items as a symbol and to generate a two-dimensional array of such symbols. A further drawback results from the the limited resolution and size of the screen. It can be difficult to keep track of the overall context in which particular items are being manipulated. For example, it may be hard to select the proper wire out of a large group of wires extending over a significant distance on the chip.

It is usually a simple matter for these systems to generate CIF, as they retain a precise description of the layout geometry. It may be difficult, however, for them to exploit the CIF symbol mechanism if the layout system has no concept of symbols or if its symbol semantics do not correspond closely with those of CIF.

7.2.4 Standard-cell and Gate-array Systems

Some IC design systems in use today free the designer from most of the details of geometric layout. The design is described in terms of gates or functional blocks and the interconnections among them. The description may be entered with an interactive graphics system, but it contains no geometric information; it is a "logic diagram." The design system processes the logic diagram, drawing on a library of gate or standard-cell geometries, and devises a layout that implements the desired logic diagram.

Ultimately, these systems generate geometric mask information, which can easily be expressed in CIF form. These systems can probably take advantage of CIF's symbol machinery to define standard cells, standard gates, etc.

7.2.5 Silicon Compilers

The ultimate goal is to interact with the computer on a much higher level, where the system designer enters a detailed functional description of his system. From there a sophisticated program, drawing upon the resources of a large collection of properly parameterized subsystems or building blocks in a library, would automatically assemble a possible layout and the corresponding CIF file.

Steps toward this goal are already being taken, see for example [Johannsen 1979] or [Ayres 1979].