HIGH-LEVEL LANGUAGE FOR AN INTERACTIVE GRAPHICS WORK STATION*

Richard Heuft and Arlen Schmidt
Electrical Engineering, University of Alberta

ABSTRACT

As the cost of colour graphics equipment continues to decrease, it will become feasible for each design engineer to have exclusive access to a dedicated interactive graphics work station. The high-level graphics language described here is intended for use in such a work station to allow a designer to enter geometric information directly into a computer and to manipulate that information interactively rather than to draw a layout by hand and have the graphics entry done remotely by an operator. In particular, the type of work station considered here is intended to aid in designing geometric layouts for Very Large Scale Integrated Circuits (VLSI).

Introduction

A wide variety of low-cost graphics work stations are becoming available which are intended for Engineering automation applications. Many of these work stations have powerful 16-bit or 32-bit microprocessors, large memories, large amounts of secondary storage, an advanced operating system such as UNIX, colour graphics capabilities and a rich set of optional graphics entry hardware such as digitizing tables, light pens, track balls, and joy sticks.

One application for such work stations is in the design of VLSI circuits. Integrated-circuit technology has progressed to the stage where it is feasible for even relatively small companies to undertake the design of custom or semi-custom. VLSI circuits to suit their individual needs. Such work stations will provide employees of small companies with design capabilities that, previously, only existed at very large corporations. In addition, Universities are beginning to teach courses in VLSI design and require work stations for design laboratories. The capabilities of these work stations are limited only by the design software available to execute on them. The proposed graphics language is intended for use. with such a work station. It will provide a circuit designer with a powerful and flexible method of laying out the fabrication masks required for VLSI circuits.

The Graphics Language

The proposed high-level language is Pascal-like and allows a designer to describe a graphical layout in a manner similar to writing a Pascal program. Features of the graphics language include:

1) graphics commands,

2) variables and expressions of the predefined types "point", "length" and "integer",

3) assignment statements,

4) symbols,

5) looping and conditional statements.

Graphics commands are associated with geometric figures such as polygons, boxes and wires. Each time a graphics command is encountered during execution of the graphics program, the specified geometric figure is drawn on the graphics output device. Variables and expressions of type "point" and "length" are used within the graphics commands to specify the coordinates of the geometric figures. These expressions give the designer complete freedom to write graphics programs to generate mask layouts. Assignment statements are used to calculate initial and intermediate values of variables.

Symbols are similar to Pascal procedures. A symbol consists of a group of statements to which the designer assigns a name. Whenever a symbol name is encountered in a graphics program, all of the statements associated with the symbol are interpreted and the geometric figure described by the symbol is drawn. Each time a symbol name is used in a graphics program, the designer can specify a transformation which will be applied to each "point" variable (and constant) in that

* This work is supported in part by grants from the National Sciences and Engineering Research Council of Canada and the University of Alberta Central Research Fund.

symbol. Consequently, symbols can be located at an arbitrary location in the layout and at an arbitrary orientation. "Point" variables defined within a symbol are particularly useful when drawing interconnection wires between points within various symbols. Looping and conditional statements are the familiar "while", "repeat", "for" and "if" statements of Pascal. Variables of type "integer" are used as the index variable in "for" statements.

Quite often in VLSI design, a particular geometric figure is repeated many times in a regular fashion. Symbols, expressions and the looping and conditional statements allow for complete flexibility in describing highly repetitive geometric structures. For example, one method of implementing an 8-input, 1-output selector circuit would be to draw eight NOR gates each of which select one of the data inputs depending on the code applied to the three selection inputs. The eight NOR gates are all similar except the NOR inputs are different combinations of the three selection input signals. With the proposed language, an individual NOR gate can be described using appropriate graphics commands. These commands can be made into a symbol and the symbol can be called eight times using a "for" statement. Each time the NOR symbol is called, a different transformation is applied. This transformation is a function of the "for"-loop index and the width of the symbol. The width of the symbol can be a "length" variable calculated by the symbol itself. "If" statements within the symbol can determine which of the selection inputs to apply to each NOR gate depending on the "for"-loop index. Also, the coordinates of the inputs, outputs and the power and ground connections for the circuit can be calculated and stored as "point" variables and will be available to other symbols which must interconnect to this circuit.

A more general example is a Programmable Logic Array (PLA). This circuit consists of two arrays of NOR gates. The first array -the AND array- is capable of having any combination of the input signals as inputs to any of the NOR gates. The second array -the OR array- is capable of having any of the outputs from the AND array as inputs to any of the NOR gates. A PLA is completely general since any Boolean function can be expressed as the sum of minterms. A PLA circuit is very fast since there are only two gate delays between input and output. PLA circuits are used quite often in VLSI designs and they are very easy to express in the proposed language.

Interactive Design

A designer using the proposed language at a graphics work station would first partition a geometric layout into a set of convenient symbols. Next, a set of strategic points would be chosen whose coordinates are either required within a symbol or are required externally by other symbols for interconnection. The designer would then begin entering each symbol using the graphics entry capabilities of the work station. A menu of the available graphics statements and language statements is presented along the edge of the display. To enter a graphics command, the designer selects the desired command from the menu and begins entering the points. If a symbol is hand drawn to scale, each point can be digitized using a graphics tablet. However, since hand drawing to scale is time consuming, the designer could elect to enter to points manually from a keyboard.

Libraries of Cells

Symbols which could potentially be used in a variety of designs can be entered once and stored in a public library to be used by many designers. Examples of widely used symbols in VLSI applications include logic elements such as the selector circuit and the PLA described above in addition to the necessary input and output bonding pads and high-fanout buffers. Once these symbols are defined and stored in a library, a designer needs only to be aware of the "point" and "length" variables which allow these symbols to be positioned in the layout and interconnected to other symbols. One trend in industrial VLSI design environments is to provide a standard set of pretested symbols which a designer can use to implement a circuit.

Summary

The proposed high-level language allows a designer to define and use symbols and variables to aid in entering geometric layouts for VLSI design. An extensive expression handling facility is provided to support both "point" and "length" variables. Symbols can be defined both to aid in structuring a given design and to enable designers to share previously-designed circuits. A work station making use of this language allows a designer to enter a layout in a natural and convenient fashion.