XIII. DIGITAL SYSTEM DESIGN AUTOMATION*

Academic and Research Staff

Prof. F. F. Lee
C. C. Hsieh
J. B. Kasarda

Graduate Students

B. S. Franklin
H-m. D. Toong

A. REGISTER-TRANSFER-LANGUAGE TO BOOLEAN-LEVEL-DESCRIPTION SYSTEM

As a part in an over-all Digital Design Automation system, a complete system design has been developed for a Register-Transfer-Language to Boolean-Level-Description system. The proposed system may be described as follows. The input to this system is a digital design description written in digital system language (DSL-1).[1] The system integrates the use of the DSL Compiler and Simulator[2] with its own procedures to produce the Boolean level description of the input design.

The Boolean level description is a hardware description of the design containing required components, component interconnections in terms of input and output signals, and the Boolean equations associated with each component. The components realizing the design will be selected from a Component Library, which is an installation file describing all available hardware components, the functions that these components can implement, and their input-output requirements. Since for any given function such as and, add, equal, etc. there will generally be many components to implement the function, the designer will have the ability to specify a specific component to be used. This will be done through Component Request cards. Otherwise a cost selection technique will be used to select a component. Functions can be implemented with a black-box operator component or through multiple functional components, as might be the case in implementing exclusive or.

The Boolean-Level-Description Generator is the main process of the system that calls upon the DSL simulator to aid in the logic generation. The DSL compiler's External Date Base is the input description of the digital design to be realized. The Boolean-Level-Description generator can achieve efficiency in the generated hardware description by sharing logic for certain operations such as add and multiply. The DSL simulator will supply the data necessary for determining whether or not

*This work was supported by the Joint Services Electronics Programs (U.S. Army, U.S. Navy, and U.S. Air Force) under Contract DA 28-043-AMC-02536(E).

logical conflicts exist in sharing a particular operator component.

The Boolean-Level-Description generator is composed of five phases: Translation, Operator Realization, Consolidation, Expansion, and Logic Editing. Translation will generate the preliminary Logic File, which is the initial logic-component description of the design. Operator Realization will determine and generate the gating logic for sharing operators, based on timing information supplied by the DSL simulator. Consolidation will search the Logic File for multiple occurrences of the same component and some input signals and replace these multiple occurrences with single components. Expansion will replace all parallel path (multiple bit register) logic descriptions in the Logic File with individual bit logic descriptions. The Logic Editor will produce hardware description documents required from the Final Logic File output of the system.

Partial implementation of the Translation phase has been carried out using the PL/I programming language to demonstrate the basic feasibility of obtaining a basic Boolean-level hardware description.

The Register-Transfer-Language to Boolean-Level-Description system is the subject of Master's thesis research.[3]

B. S. Franklin

References

1. F. F. Lee, "Machine-Aided Digital System Design Language," Quarterly Progress Report No. 97, Research Laboratory of Electronics, M.I.T., April 15, 1970, pp. 193-200.

2. J. B. Kasarda, "A Machine-Aided Digital Systems Design Language," S.B. Thesis, Department of Electrical Engineering, M.I.T., February 1970.

3. B. S. Franklin, "A Register-Transfer-Language to Boolean-Level-Description System," S.M. Thesis, Department of Electrical Engineering, M.I.T., September 1970.