VLSI design shrinks to mere man-weeks

Researchers at the Massachusetts Institute of Technology's Lincoln Laboratory say they have come up with an automated circuit-design tool that will dramatically cut the time currently required for designing and laying out high-performance custom very large-scale integrated chips.

"Microprocessors in the Motorola MC68000 or Zilog Z8000 class take an average of 20 man-years of design and layout time," observes research team leader Jeffrey M. Siskind at the Lexington, Mass., facility. "We expect our system to do a comparable job in a few man-weeks."

The MacPitts design system speeds things up by intervening much earlier in the design process than do current design tools, Siskind explains. It accepts high-level, algorithmic descriptions of integrated-circuit functions and generates from these the final mask-level geometric layout for an actual circuit. By contrast, computer-aided design systems

and special layout languages used both commercially and in many universities typically automate only the last, lowest-level phase of design, the translation of gate-level diagrams into mask specifications.

The MacPitts project is similar to the one at the California Institute of Technology where Carver Mead hopes to develop a silicon compiler—a program that transforms circuit descriptions into a set of masks. Currently, Siskind says, the chip designer is faced with the "time-consuming and very error-prone chore" of refining an initial functional plan down through the architectural, gate, and transistor levels.

The designer also must check for design errors each step of the way, tying up hours of computing time running error-checking and node-extraction routines or switch-level simulations on transistor schematics extracted from a hand-done layout.

Few pages. Working with the MacPitts system's design language, however, the designer can sum up a fairly complex signal-processing or similar custom circuit in a few program pages, Siskind says. Based on a standard design language called Lisp, the MacPitts language contains semantics that let it describe concurrent processes usually required in high-throughput custom ICs. Other features, like multiple-way branching and nested subroutine calls, help keep programs short.

Siskind reports he and colleagues Jay R. Southard and Kenneth W. Crouch have fully described 20,000-transistor ICs using programs running only four pages (about 200 lines of code) or less. Complex VLSI circuits of a few years hence could be captured in something like 10 to 15 program pages, he estimates.

A system compiler reads the source program descriptions and translates them into a target-architecture organization customized to the particular IC functions specified. The target architecture partitions systems into two sections—a data path and a control unit.

The data path is described as sets of primitive units—register arrays interspersed with functional and test

Electronics review

units like adders, shifters, and comparators—connected on parallel buses for concurrent operations. The control unit, in charge of governing the data path's sequence of performed operations, is described in terms of Boolean algebra.

Once higher-level compiler routines have assembled this intermediate-level architectural description of the IC, lower-level routines can use the description in generating mask-layout patterns for circuit fabrication. At present, the MacPitts system's compiler outputs descriptions in CIF, a code geared to n-channel MOS production. But the intermediate-level description itself is technology-independent, notes Siskind. Thus, substituting different mask-generating compiler routines, one could adapt the same IC description to complementary-MOS or some other IC technology.

The intermediate-level description also drives a functional simulator, a compiler program that emulates the logical behavior of the IC at the architectural level. The simulator is equipped with semantic-error-checking capabilities, aiding in verifying design correctness.

Error-free layouts. Such a verification process will take less than half an hour, according to Siskind, and it need occur only at the end of the design process, just prior to fabrication. Because the MacPitts compiler automatically constructs circuit descriptions in accordance with built-in design rules and tolerances, it produces virtually error-free layouts, he asserts. This exactness makes unnecessary repeated iterations of routines like design-rule checking and node extraction that are required for hand-executed designs.

Work on the MacPitts system, described by Siskind and his colleagues at a conference on advanced research in VLSI held last month at MIT in Cambridge, Mass., should culminate by mid-year. By June, Siskind hopes to be fabricating n-MOS devices with the system. After that, MacPitts will come into the public domain with the publication of a summary report and system documentation, he says. **-Linda Lowe**