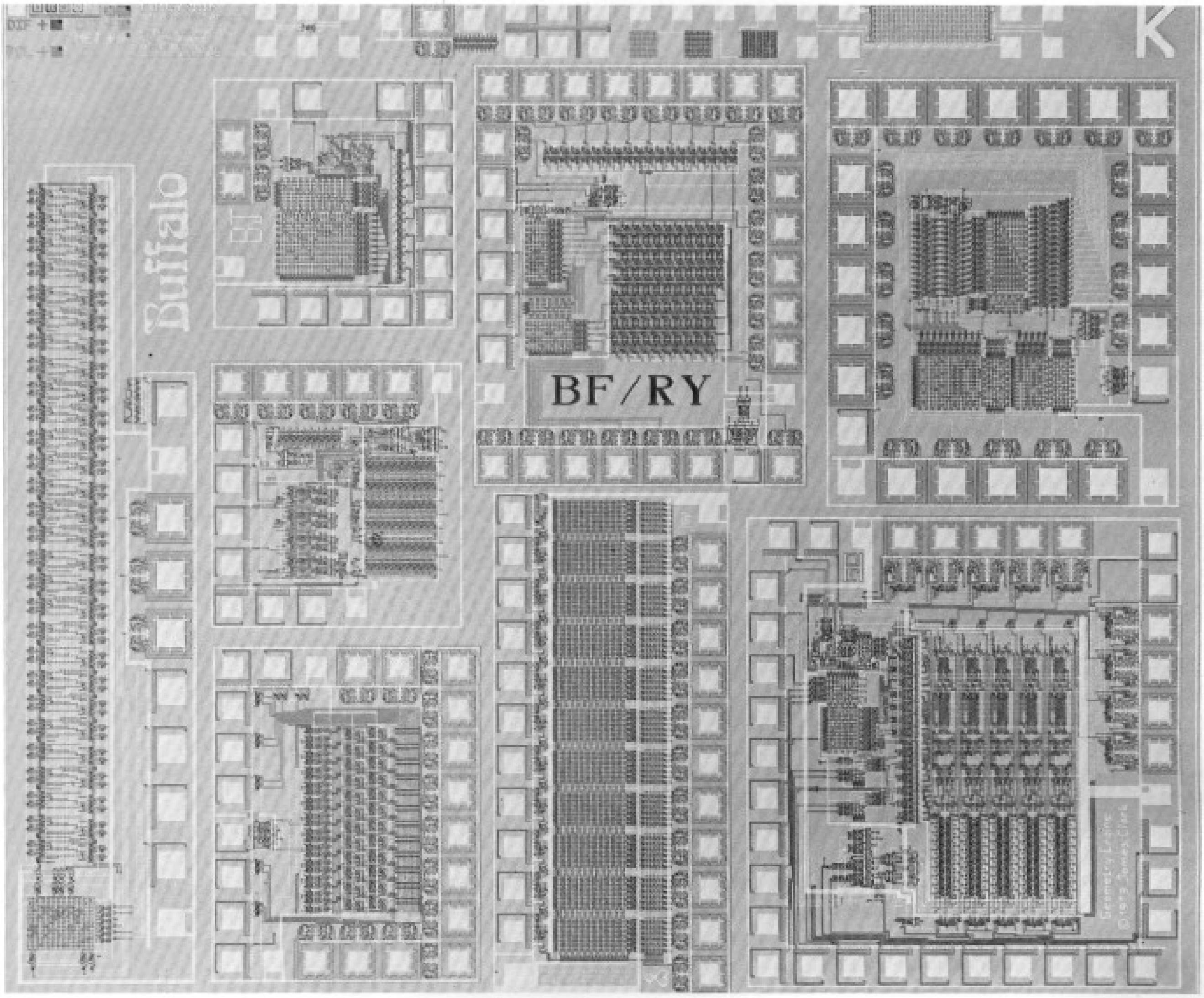
MPC79

A large-scale demonstration of a new way to create systems in silicon

Lynn Conway, *Alan Bell*, and *Martin E. Newell*, Xerox Palo Alto Research Center

Photomicrograph of one of the MPC79 Project Chips (Die-type K, Stanford University)



Melgar Photography

Computer architects and system designers usually create experimental systems and design prototypes of new products by using off-the-shelf TTL parts. System designers are familiar with these parts and with the procedures for building systems out of them.

As we'll see, the cost and time to design and prototype with TTL parts are very much less than traditionally required for mapping the same function into a custom LSI chip. Therefore, up to now few system designers have been directly involved in the design of integrated circuits. IC design has been left to circuit designers, and is considered by system designers to be a rather mysterious, difficult craft. The circuit designers are called on only when a design has an enormous potential market, and the high front-end IC design costs can be amortized over a long product run. Of course, once a design has been mapped into silicon, the cost of copies is far lower than would be the equivalent in TTL.

But is it necessary for custom LSI design and prototype implementation to cost so much more than the same thing in TTL? The answer turns out to be NO! A simplified method of integrated-system design has been developed that is as easy for the system designer to learn and use as is TTL design. In addition, a new type of "VLSI implementation system" has been developed that brings the time and cost of implementing prototypes of a chip's design down to that of implementing the same function in TTL.

This article first contrasts the costs of TTL versus traditional MOS-LSI design. Some interesting order-of-magnitude results are brought out, even using very rough comparisons. A sketch is given of the activities that led to the new integrated-system design methods. With this as background, the article describes the development of the new VLSI implementation system, which permits remote-entry, fast-turnaround, low-cost implementation of many individual design projects in each maskmaking and wafer-fabrication run.

During the fall of 1979, the LSI Systems Area at Xerox PARC organized a major test of the *entire set* of new VLSI design and implementation concepts. In an effort known as the "MPC79" project, we conducted the demonstration-operation of the VLSI implementation system, servicing a user community composed mainly of students taking courses in VLSI design at universities throughout the United States (Conway, Bell, and Newell January 1980). A total of 82 design projects, created by 124 designers, were implemented in the resulting multiproject chip-set. Implementation began 4 December 1979. Packaged chips, custom wire-bonded for each project, were distributed to the designers on 2 January 1980. The average effective implementation cost was less than $500 per project.

The MPC79 project, and the university design activities it supported, provide the "script" for a new way to "do electronics." Those system designers and companies who follow this script can directly design and prototype in VLSI with the same ease of design and low prototyping costs normally associated with TTL design. The resulting designs have tremendous advantages of high performance, low power dissipation, small size, and very low cost per copy relative to their equivalents in TTL. In addition, we're seeing a wave of discovery and invention, and the production of novel systems not feasible using TTL, as designers begin to uncover the true architectural possibilities of VLSI.

Prototype Design and Implementation Using TTL

Suppose you want to prototype a new digital subsystem. Perhaps you're a designer in a computer manufacturing firm, and are working on a novel disk controller or CRT display controller. Or you are conducting research in digital system architecture, and want to prototype a special purpose subsystem and interface it to your computing equipment in order to conduct some real-time experiments.

At present, you'll likely think of designing by using TTL. Even if you use a microprocessor or other off-the-shelf LSI, you'll still be faced with creating the custom portion of your design using MSI-TTL, proceeding as follows. After blocking out the basic architecture of your subsystem, you refer to catalogs of TTL parts to select the appropriate registers, counters, gates, etc. You then complete the detailed logic design to produce a schematic of interconnected parts corresponding to the starting architecture. Finally, you sketch the placement of these parts into perhaps one wire-wrap board and prepare a list of the appropriate wiring interconnections. The result is a specification of the design, ready for prototype implementation. You might do this with- out computer aids, or you might use some computer aids to assist in board layout and wirelisting; in either case, the task is conceptually much the same. To implement your prototype, you first order the necessary parts. The circuit board is wire-wrapped, and the TTL chips (when available) are inserted into the board. The subsystem is now ready for basic functional testing.

Now, how long does this take, and how much does it cost? One person can design a board full of MSI-TTL in one to two months. So, let's say the design takes six weeks, and in effect costs about $6000 (counting overhead). Implementation of the prototype will take two to six weeks (remember, you have to order the parts), and the parts and labor costs will total about $500. Therefore, to get the first working subsystem takes about 10 weeks and $7000. Additional copies then require two to four weeks' time and cost $500 each.

Present-Day Design and Implementation in MOS-LSI

Suppose we wanted to design and implement a "TTL board-equivalent" size subsystem in an integrated system technology such as MOS-LSI. How long do you think it would take, and how much would it cost, to get to the first working prototype? Let's examine how it would be done using present practices in a semiconductor firm.

First, an MSI-TTL prototype might be made, as above, to test the functional behavior of the subsystem in the target environment. If so, that will take 10 weeks and $7000. Now the real work begins. The task is re-transferred at the architectural level into a hierarchical team consisting of (1) a logic designer who maps the design into an appropriate logic design for MOS-LSI, (2) circuit designer(s) who remap the logic design into a circuit specification, and (3) layout designer(s) who transform the circuit design into the layout of patterns to be printed on silicon. The layout, or portions thereof, is hand drawn and entered into a computer system by an operator using a digitizing table. There will be many iterations as the team works out interactions between levels of the design. It will take this team six months to a year to produce a final layout, ready for the first attempt at implementation. There will be an average of two or three people working on the design at any one time. The preparation of the design for first implementation may cost roughly $100,000.

In order to implement chip prototypes, the layout description is converted to pattern generator format, and a set of masks made by a mask-making firm. This takes three to six weeks to set up and complete, and will cost about $6000. Then a short run of wafers (one boat load of 24) will be queued to run on a wafer fabrication line. This will take another three to six weeks, and effectively costs at least $4000. The packaging of some chips will then take a week or so. Thus, the implementation from design file to packaged chips takes about three months, and the direct cost is about $12,000. There are presently no industry-standard interfaces between design and implementation. Thus the setup of implementation can involve many people, with forms to be filled out, procedures to be planned and coordinated, etc., leading to large implementation "front-end" costs. We estimate these costs to equal a moderate fraction of the direct costs of mask-making and a minimum wafer fabrication run.

After the prototype chips are tested, there are usually one or two redesign/reimplement iterations to fix design and layout bugs and to optimize the design further. Each of these iterations will cost perhaps another $5000 for design work, and the direct costs of $12,000 for implementation. The redesign might take a month, and the reimplementation take three months.

Therefore, the overall cost of the LSI chip design and implementation will be roughly $150,000, and will take about one to 1-1/2 years! However, once the masks for a working design are available, copies of that part can be produced in quantity for just a few dollars apiece, since the masks for a working design can be used to print thousands of chips, even in a minimum wafer fabrication run.

We see that there is a very great contrast between the time and cost of designing and implementing a typical subsystem in TTL as opposed to LSI: One person can do it in TTL, in 1/5 year, for a cost of $7000. However, it takes a team to do it in LSI, takes 1 to 1-1/2 years and costs $150K. There's an even greater reverse contrast between the cost of additional copies of the subsystem in TTL versus LSI: The TTL copies cost $500. But extra LSI copies cost only a few dollars each. So the moral (so far) is: If you're only going to make one, or a modest number, do it in TTL. If you have a very large market for a subsystem, do it in an integrated system technology such as MOS-LSI.

There is a world of potential applications, each one requiring modest numbers of systems, going unfulfilled because the cost of TTL implementations is too high. Although the cost per copy of LSI implementation is extremely low, the present front-end costs for LSI design and implementation are too high for such applications.

The status quo of high LSI front-end costs is maintained by design efforts focused on the production of competitive products for known markets. Such efforts necessarily involve lots of optimization. These optimization aspects of LSI design then promote the notion that such design is complicated, hard to do, and a lot of hack-work. Thus, few designers think of prototyping or doing exploratory design work directly in LSI/VLSI, and as a result, very few system designers now even know how to design in LSI/VLSI.

Developing a New Way to *Design* Integrated Systems

The LSI Systems Area at Xerox PARC and the Caltech Computer Science Department have collaborated for a number of years on the exploratory development of new VLSI design methodologies. This work is aimed at simplifying integrated-system design so that it can be learned more rapidly and practiced more widely by system designers than in the past. As new design techniques have been uncovered, they've then been tried out and debugged in actual practice, both in the university classroom and during in-house short courses at PARC.

This has been a process of discovery and iteration. Proposed design techniques have been taught to university EE/CS students and to practicing EEs and computer scientists, with the students then doing integrated-system design projects as part of their course experience. Groups of these projects, and thus the design methodology itself, have been subjected to the acid test of actual implementation and testing. The technology used in this activity is nMOS, chosen because of its potential for conceptual simplicity of design, the high density and performance of present nMOS processes, the clear future for further scaling improvements, and the ready availability of access to nMOS wafer fabrication.

There has been success in this research, leading to the publication of the textbook *Introduction to VLSI Systems* (Mead and Conway 1980). This text describes new techniques that enable system designers to design directly in integrated form just about as easily, rapidly, and economically as when using off-the-shelf components. A number of major universities, including MIT, CMU, Stanford, Caltech, and UC Berkeley, have begun to offer standard courses in VLSI design based on this text. The students taking these courses are mostly seniors or first-year graduate students of EE/CS having course backgrounds in digital design, computer architecture, and computer programming. Course background in such areas as analog circuit design, device physics, and integrated circuit processing technology is not a requirement for participation.

The courses proceed as follows: The first half of the course, lasting about six weeks, covers all the basics of integrated system design necessary to undertake a project. The students then innovate a project concept, and carry out its architecture, design, and layout during the second half of the course. Layout descriptions are typically produced by encoding in a symbolic layout language. Such languages provide an easy way to create, edit, plot, file, and transmit layout descriptions, even in an environment having modest computing facilities. In some schools (for example MIT), analysis aids such as circuit extractors and switch-level simulators are being used to validate the designs. The use of such aids can prevent design bugs that would require an implementation iteration, serving a function similar to the basic initial board testing and patching done following TTL implemention.

The projects produced in the six-week project phase are often of the complexity of a "TTL board equivalent" subsystem. The courses thus demonstrate that, by using the new techniques, the cost and time to design a prototype subsystem directly in integrated form are roughly the same as it would be in TTL.

Ah, but now what about implementation? Remember, using standard industry practices it takes about three months and $12,000, plus whatever costs are involved in setting up the interfaces and handling the logistics, to turn a layout design file into packaged chips. That's a long time and a lot of money for the implementation of a student project!

Developing a New Way to *Implement* Integrated Systems

Believing strongly in the importance of learning by doing, Prof. Carver Mead, in early integrated circuit design courses at Caltech, had his students carry out design projects, and then arranged for implementation. To share the overhead involved, the projects for a group of students were packed together and simultaneously implemented as one chip-type. Thus the notion of "multiproject chip" emerged, as a way of conducting a "group tour" through implementation, with only one person having to know the way. A group of 10 or 20 students at a time could share the implementation costs, resulting in an effective direct cost on the order of $1000 per project. That's still a lot for a student project, but mask and fabrication services were usually donated by industry. Turnaround times were rather long and unpredictable, and there were the hidden front-end costs implied by the need for an expert "tour guide." However, projects were being implemented and this proved to be highly motivating to the students.

During the collaborative Xerox/Caltech work to develop, debug, and document simplified methods of integrated-system design, a parallel effort was conducted to simplify implementation procedures. This was needed to quickly test the new design methods to see if designers could learn rapidly, design rapidly, and actually create things that worked. Standards were developed (for nondesign layout artifacts such as scribe-line profiles, alignment marks, etc.) that proved acceptable to different mask and fabrication lines, simplifying setup procedures and reducing front-end costs. A simple ratioed, scalable set of nMOS design rules, developed as part of the new design methodology, also helped to simplify the design/implementation interface. A standard design-interchange format (CIF2.0) was developed that is machine-independent, has simple and unambiguous semantics, and provides compact representations of layout geometries. CIF2.0 enables designers at different locations to communicate design files easily and share various facilities. This effort and the results are documented in the *Guide to LSI Implementation* (Hon and Sequin March 1980) .

In the fall of 1978, in a test of the transportability of the project-oriented courses, the author introduced the first VLSI design course to be offered at MIT. The CIF2.0 design files for the 19 projects from that course were transmitted over the ARPANET from MIT to Xerox/PARC, where they were merged and converted into a multiproject chip set mask-specification, and then rapidly implemented in a joint effort by Xerox, Micro Mask, and HP-ICPL. Six weeks following the file transmissions to PARC, packaged project chips were returned to the students. Several of the projects worked completely correctly. This effort demonstrated not only course transportability, but also that the new standards enabled remote-entry and very fast turnaround of a moderate number of projects, even from a community of new designers at a new location. Although the effective direct implementation cost per project was still on the order of $1000, the projects were fairly large in scope (typically being the equivalent of a boardful of TTL).

The Idea of a "VLSI Implementation System"

By early 1979 we at PARC were in contact with many universities that were preparing to offer courses similar to the MIT 1978 course. Instructors at these schools all wished to obtain rapid implementation of student projects. It became clear that a demand for fast-turnaround implementation was developing in the universities that was far beyond our ability to supply using our earlier methods. Those methods, requiring "expert guides," could not be scaled up to a large user community without a corresponding scaling up of the front-end costs. We perceived the need for simple and standardized dynamic procedures for user interactions with implementation, in addition to the "static" design/implementation interface standards.

We then thought of creating something analogous to a "time-shared operating system," to interface many remote users in a standardized way with centralized maskmaking, wafer fabrication, and packaging facilities. Such a system could provide informational and instructional messages to users, make automated responses to user messages, juggle the many space and time constraints involved in implementation, and assist in the management of the overall effort. We believed such a system could greatly reduce implementation turnaround time, could bring access to the fabrication process to a larger, more dispersed technical community than in the past, and could further reduce the implementation cost per design (especially the front-end cost). We decided to evolve such a system using the method previously applied to develop the design methodology: by design, use, and iterative debugging in the universities, with the system being experimentally operated to provide an implementation service to university designers.

Designing and Building the MPC System

Some important factors converged to enable construction and demonstration of such a system. First, almost all the major universities now involved were on the ARPANET, and could interact in a uniform way with a remote system via electronic messages and file transfers. The computer science community in these universities is very familiar with the idea of widely scattered groups of people being involved in "network adventures" of various sorts, and such activities are common within this culture (The November 1978 edition of the *Proceedings of the IEEE* (Kahn, 1978) is an excellent source of information about such computer-communications networks). Therefore, there were enough potential users to enable a demonstration on a scale that would provide a thorough test of a system. Next, the effort by Hon and Sequin had by then produced a well documented set of the necessary static standards. Finally, we had just been among the first commercial users (for the MIT project set) of Micro Mask's new ETEC electron-beam maskmaking system. In addition to permitting fast-turnaround, that system enables more die-types to be merged together per mask set than is feasible with optical pattern generation, but without significant increase in cost per mask. Thus, we could extend the multiproject chip idea by merging *multiple* multiproject chips per wafer, achieving additional economies in the prototype implementation cost per project. And so, we decided to launch the MPC79 project.

The architecture of a prototype multiproject chip (MPC) implementation system was undertaken and

Principal functional modules of the MPC system:

The *message and design-file handler* provides an "electronic mailbox" interface between the MPC system and the user community. It is basically a network message handling facility tailored to the needs of an implementation service. It also has the capability of retrieving design files referred to in messages, and running them through a CIF parser for syntax checking and determination of bounding box sizes. Although the MPC system processed the request messages under the control of an operator during MPC79, it was designed to simulate the functions of a fully automatic system. To this end, all communication with users are conducted using electronic mail, and the messages are required to be in a highly constrained format for specific requests, making heavy use of keywords (e.g., REQUEST OPEN to open a new account, REQUEST IMPLEMENT to place a project in the queue of those to be implemented). Much of the operation of this handler is, in fact, automated. For example, the checking of CIF files, the construction of an accept or reject message based on that check, and the sending of the response message are all fully automatic. The message handler maintains the primary data base for the implementation system, from which status reports are generated and transmitted to the users. After all requests have been processed for a project set (for example, a mask set is filled, or a predetermined design cutoff time is passed), the data base is passed on to the die-layout planner.

The *die-layout planner* takes the bounding box size specifications of the projects, and packs them into dice of sizes appropriate for later packaging, under the constraint of not allowing projects to overlap. In MPC79, we wished to apply other constraints on the location of projects such as placing the designs from each university on the same die when possible. Some constraints were not definable at the time we were building the system (e.g., how we would select projects for inclusion if the set were overbooked), so we chose not to implement a totally automatic packing program. We instead built a highly interactive, graphics-oriented system that presents the operator a view of the dice and the projects. Projects can be easily moved around and packed by the operator, and the interface presents a menu of project ID's, marked-up to indicate the projects successfully packed. The output from the die-layout planner is a *planning document* that is passed to the next phase, the *file-merger/MEBES-converter*. The planning document (printable in human readable form), together with the individual CIF design files, completely defines the entire chip set. It contains such information as which projects go where on which dice, the definition of layer codes for each project, which dice go where on which wafer, parameters for conversion to MEBES format, etc.

The *file-merger/MEBES-converter* takes the planning document file and the referenced CIF files and carries out the task of merging the files together into the specified die-types, and generating the necessary MEBES mask-layer files for the electron-beam maskmaking system. Full CIF2.0 is supported by the converter, a facility that paid off since *all* capabilities of CIF2.0 were used by the MPC79 project set. The converter is fully automatic, and can be run in parallel on several machines, up to one machine per die-type. This is done by sending the planning document to each participating machine, along with a list of the dice to be converted. The converter then selectively executes the relevant parts of the planning document, and transmits its results to a central file server. The final transfer of these files to the magtape for transfer to maskmaking is carried out under control of a program that is automatically generated from the planning document, and so provides an "end-around" check that all necessary files have indeed been generated and transmitted to the file server.

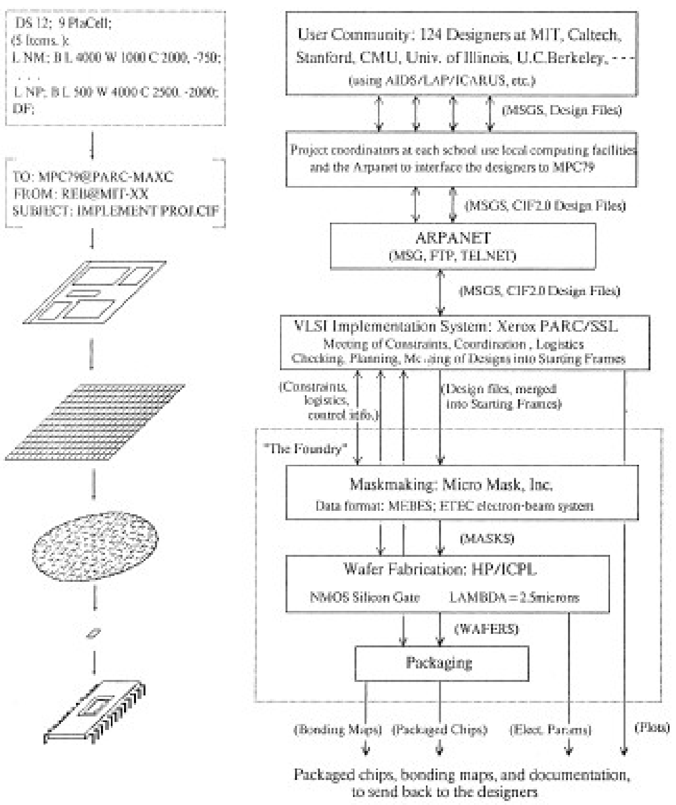
completed during the summer of 1979 by Alan Bell and Martin Newell. The system they configured has three principal functional modules (see insert): (1) the *message and design-file handler*, (2) the *die-layout planner*, and (3) the *file-merger/MEBES-converter*. Under the control of an operator, who interacts with the system using a high-level graphics interface, these modules interact with the user community via electronic messages and design file transfers, and build a data base of design files that are then merged and converted to create the multiple multiproject chip mask specification. The system produces additional information associated with the multiple projects (for example starting frame maps and wire-bonding maps), to be conveyed with the masks to the later processing and packaging steps of implementation, and back to the designers with their packaged chips.

In a crash effort, Alan Bell and Martin Newell completed the design and coding of the MPC system during the fall of 1979. By early fall, the MPC system effort was far enough along for us to make a commitment to the universities to implement projects for the fall courses. The message and design-file handler was ready by the announced date for submission of REQUEST messages. The remainder of the system was completed by the design cutoff date.

Running the MPC System for MPC79

Shortly after the courses began, the MPC79 service was announced over the ARPANET to faculty members and researchers whom we thought might be interested. A *project coordinator* was selected at each participating university to supervise the university's interactions with the MPC system. MPC79 informational messages were transmitted periodically to inform participants of schedules and deadlines, space allocation guidelines, how to get library cells, and to provide users with a guide to the constrained command-message form of interaction with the MPC system.

FIGURE 1. Flowchart of the MPC79 activities, and a view of the corresponding sequence of artifacts.



```
DS 12; 9 PlaCell;
(5 Items. );
L NM; B L 4000 W 1000 C 2000, -750;
 . . .
L NP; B L 500 W 4000 C 2500, -2000;
DF;
```

```
TO: MPC79@PARC-MAXC
FROM: REB@MIT-XX
SUBJECT: IMPLEMENT PROJ.CIF
```

User Community: 124 Designers at MIT, Caltech, Stanford, CMU, Univ. of Illinois, U.C.Berkeley, - - -
(using AIDS/LAP/ICARUS, etc.)
(MSGS, Design Files)
Project coordinators at each school use local computing facilities and the Arpanet to interface the designers to MPC79
(MSGS, CIF2.0 Design Files)
ARPANET
(MSG, FTP, TELNET)
(MSGS, CIF2.0 Design Files)
VLSI Implementation System: Xerox PARC/SSL
Meeting of Constraints, Coordination , Logistics
Checking, Planning, Me .ging of Designs into Starting Frames
(Constraints, logistics, control info.)
(Design files, merged into Starting Frames)
"The Foundry"
Maskmaking: Micro Mask, Inc.
Data format: MEBES; ETEC electron-beam system
(MASKS)
Wafer Fabrication: HP/ICPL
NMOS Silicon Gate
LAMBDA = 2.5microns
(WAFERS)
Packaging
(Bonding Maps)
(Packaged Chips)
(Elect. Params)
(Plots)
Packaged chips, bonding maps, and documentation, to send back to the designers

The courses were run on a tight schedule to correlate with the MPC79 schedule. The courses began in mid-September, and by early November, students had learned enough about the basics of design to originate a project concept and begin design, working towards a cutoff date of 4 December. Thus, students learned how to design in six or seven weeks, and completed a project in the remaining six or seven weeks of the semester. University researchers were also given access to MPC79, leading to some ambitious, team projects by researchers using sophisticated design aids of their own creation (Holloway et. al. January 1980).

The organizations we had worked with to do the 1978 MIT project set worked with us again to do MPC79. Data communications were supported using the ARPANET. E-beam maskmaking was scheduled to be done by Micro Mask, Inc. With the support and encouragement of Merrill Brooksby and Pat Castro, Hewlett-Packard agreed again to donate wafer fabrication. The fabrication was scheduled to be run at the HP Integrated Circuit Processing Laboratory managed by Pat Castro. The flow of MPC79 information and artifacts through these organizations can be visualized by studying the flowchart in Figure 1.

A variety of different design systems were used by the designers. The different schools had their own "home-brew" design aids running on different computer systems, the only constraint being that the design systems had to generate CIF2.0 layout specifications. As designs neared completion, the coordinators gathered up design files from the students and transmitted them to the MPC system at PARC, using the ARPANET to send the CIF files to reserved areas on a computer at PARC and to send network messages containing service requests to MPC79@PARC-MAXC. Requests were processed, and accept or reject messages, along with diagnostic information, transmitted back to the coordinators.

We began receiving OPEN requests shortly after 8 November, the date the service was placed on the network. Request activity increased steadily in the ensuing weeks. The volume of network activity generated was prodigious, the overall number of messages totaling about 650. Many of these messages had design-file transmissions associated with them. As one might expect, this activity reached a peak in the final hours before the cutoff time of 5:00 pm on 4 December. Indeed, almost all of the projects were resubmitted from around the country during the last day (to include refinements, correct errors, etc.). We did not censor designs. If a design met certain prearranged requirements for space allocation, consisted of syntactically correct CIF code, and was submitted by the coordinators as "finished" before the deadline, then it was included.

In a few cases, designers took on a bit too much for the available time. There were cases where last-minute panic changes led to glorious disasters. But that was all part of the happening!

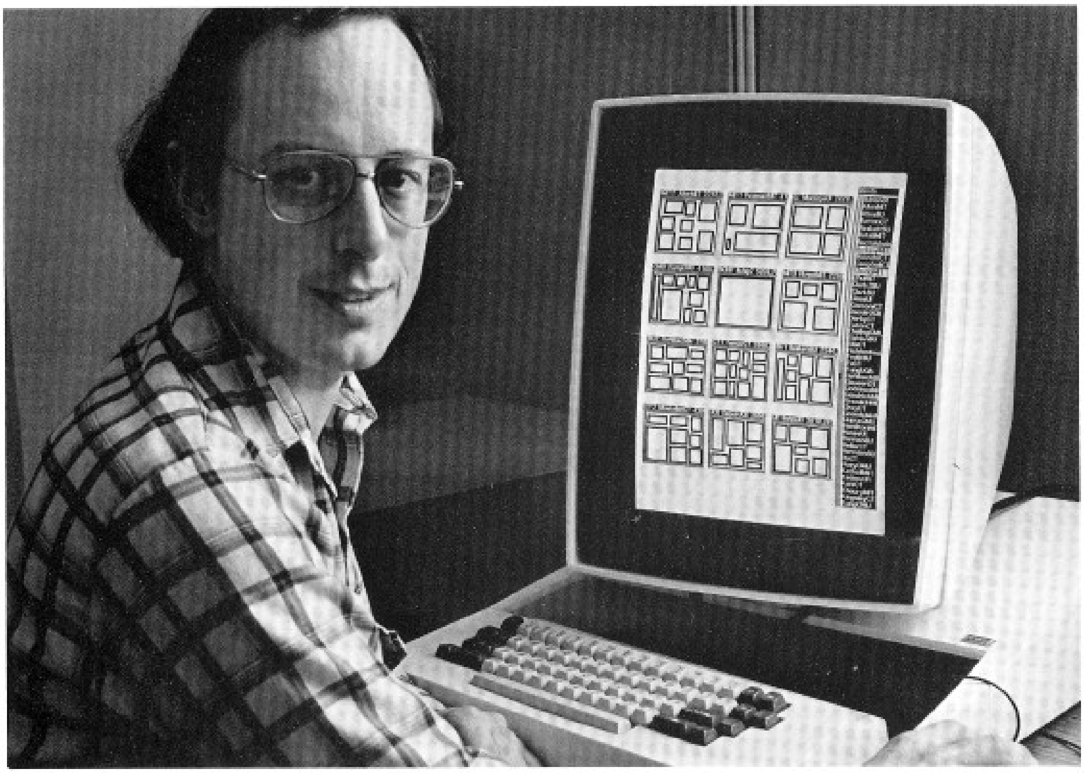
At the cutoff time, we shut down the external network communications, and began final processing of queued requests. After all requests had been processed, we summarized the mask area required by each university, and determined the number of dice to be allocated to each. An assignment of the resulting 12 die-types to die-positions within two masks sets was then completed. The die-layout planner was then run and the detailed assignment and positioning of projects was carried out (see Figure 2). Where possible, we tried to place projects from only one university within each die-type. It took two hours to achieve the final satisfactory layout.

Once we had established where and on which die each project would go, the actual merging and conversion to MEBES format was begun. This phase lasted some four hours, with the system running in parallel on three large computers. MEBES plots of the metal layer of each of the 12 die types were generated as a final visual check of the whole merge and conversion process. Early on the morning of the 5 December, we took the merged mask-specification data to Micro Mask.

The first masks of both mask sets were finished by the afternoon the following day. By then, the group at Hewlett-Packard/ICPL had already started the processing and were ready for the first masks. Generation of the remaining masks was pipelined with fabrication during the next few days. Processing continued normally except for one major contingency: the failure of a poly-deposition system (causing a delay of about eight days for repairs).

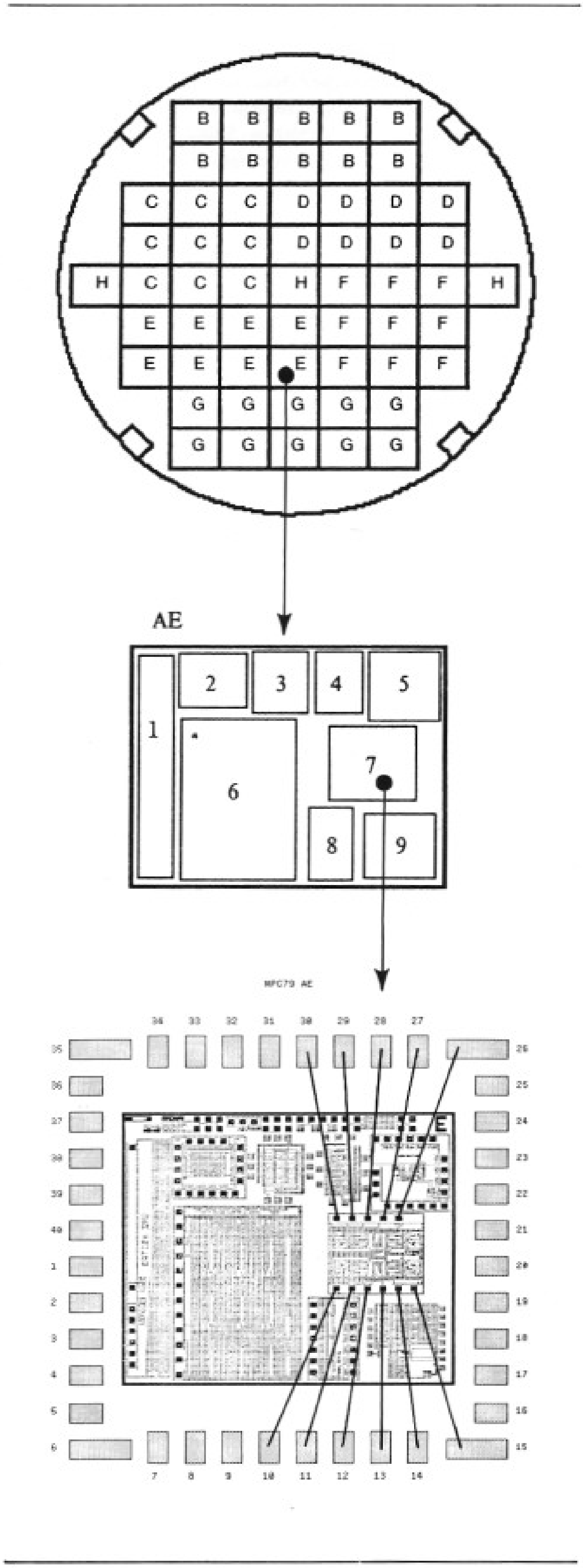
On 28 December, shortly after the first wafers began to come off the line at HP-ICPL, Jim Clark's "Hourglass" project was tested. It proved to function completely correctly, indicating the overall implementation effort had been successful. We scribed and diced the wafers, and mounted individual die into standard 40-pin packages, mounting enough to provide two or three packaged chips per project. Projects were then wire-bonded, and a wire-bonding map produced for each project to show that project's pinout. After a final inspection, the packaged chips were then boxed for shipment.

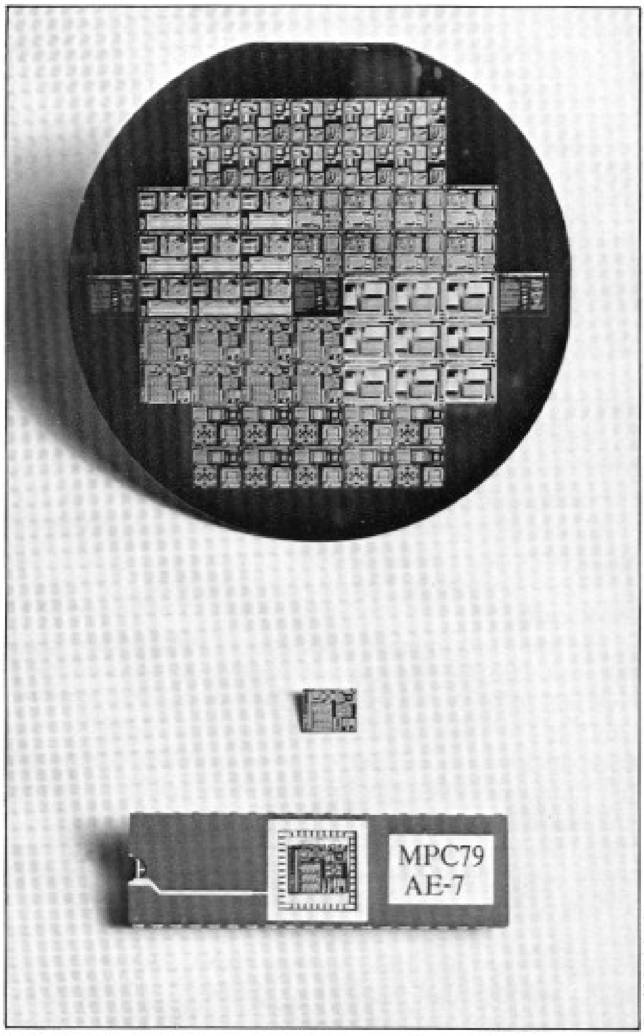
The packaging and distribution of the chips, while a seemingly mundane task, did involve considerable information management. Remember, many projects had been merged together into the mask and fabrication run. Maskmaking and fabrication, being pattern independent, are not complicated by this fact. However, at packaging time we have to reverse the process of merging, and get the right chips and data back to the right designers. Each MPC79 project was given a unique Wafer-type/Die-type/Project-number code, as a function of its location on the wafers. For example, Project number 7, on Die-type E, from Wafer-type A, is labelled "AE-7" (see Figure 3). Each packaged chip was labeled with the code of the project within the chip that was bonded to the package leads. The chip package, the project's wire-bonding map, and also the plastic box containing the packaged chip, were all labeled with this code.



David Bickford

FIGURE 2. Alan Bell interacting with the MPC system to construct the MPC79 die-layout plan.





David Bickford

FIGURE 3. At Right: Photo of MPC79 type-A wafer, type-AE die, type-AE-7 packaged chip.
At Left: Corresponding hierarchy of informational material.

A documentation effort was run in parallel with the packaging. Many university participants had never seen or handled chips before. Thus we created a document for participants that described how to identify their project's code, how to figure out pinouts using custom wire-bonding maps and package data, how to prepare chips for testing, etc. This "Implementation Documentation" (Conway et. al. January 1980) also contained information about the starting frame, and data from HP-ICPL concerning the electrical parameters of the process, permitting estimation of minimum clock periods, etc. Most of this documentation was produced directly from the design-file data-base and archived records of the MPC79 die-layout-planning/design-merging process. This method of rapid document creation enabled us to return that material in a timely way to the designers, right along with their packaged chips. The packaged chips, the wire-bonding maps, and the Implementation Documentation were shipped out to all participants on 2 January, 1980.

The Results

All the university courses were able to stay synchronized with the MPC79 schedule, and most of the students were able to complete projects for inclusion in MPC79. This was a remarkable accomplishment at certain of the universities where courses were being offered for the first time, and where design aids were being programmed and debugged in parallel with the courses.

The resulting chip set contains a total of 82 integrated system design projects from 124 participating designers. Designs were included from courses at MIT, Caltech, Stanford, University of Illinois, and University of Rochester (see also the *University Scene* column in this issue). MPC79 also includes a number of designs by faculty and research staff members at MIT, CMU, Stanford, UC Berkeley, University of Bristol (UK), University of Washington, Yale University, and University of Colorado (Colorado Springs). Most designs were communicated over the ARPANET; a few were sent over other networks. The implementation turnaround-time, from design cutoff to distribution of packaged chips, was 29 days.

Many of the projects have already been functionally tested, and results reported in feedback questionnaires returned by the participants. Some major projects such as a 6mm x 7.5mm LISP microprocessor have proven to function correctly. Most designers discovered simple design errors of the sort one makes when first creating a large program in a new language, and which can be corrected by another iteration. Some designers at MIT had access to advanced analysis aids such as circuit extractors and switch-level simulators, and these aids enabled the users to avoid almost all such errors (Holloway et. al. January 1980). Yield per project appears to be high: of the large LISP-machine chips tested so far, roughly one quarter function correctly. Thus, small projects will have very high yields.

The chip set was implemented as 12 different project die-types on two different wafer types. The effective average cost per project is estimated as follows: The masks cost about $7500 per set; one boatload of wafers per mask set, generating enough chips for prototype testing, effectively cost perhaps $4000 to $6000 (two boatloads were run per mask set, to protect against contingencies); thus the effective cost of maskmaking and fabrication was $25,000. To this must be added the costs of data communications, computing, packaging, and operation of the MPC system. A conservative (high) estimate of these last costs might be $10,000 to $15,000 (counting those operation costs as would occur in an ongoing use of the system). Thus, the 82 projects effectively cost $35,000 to $40,000 to implement, for an average of $400 to $500 per project. The frontispiece shows a group of average sized projects. As can be seen, the projects are of substantial scope, each being roughly on the order of a "TTL board-equivalent" subsystem.

Therefore, we have demonstrated, on a large scale, that designers can design moderate sized integrated systems, and that prototypes of such integrated system designs can be implemented, all at a cost in time and money roughly equal to that when using off-the-shelf TTL parts.

As a result, it is now possible for larger numbers of systems designers to explore more freely the limits of what is possible using VLSI technology. In fact, on examining the MPC79 project set, we find many new applications are being explored, new architectural techniques discovered, and novel mappings of logic functions into nMOS cells are being invented by university participants. New and powerful design aids have been created and tested during efforts to create some of the projects. Interesting examples of such activities have already been reported (Clark January 1980 and Holloway et. al, January 1980), and we look forward to results of further university activity in this area.

A final note about our methods in achieving these results: The research of the LSI Systems Area has often involved the experimental introduction and debugging of new technical and procedural techniques by using the computer networks to interact with students and researchers in the universities. This methodology was applied on a very large scale in the MPC79 project. There are risks involved in using such methods: the risk of failure and the risks associated with presenting undebugged technology and techniques to a large group of students. However, we have found the universities eager to collaborate and to run these risks with us, and we look forward to a continuation of such activities. It is exciting and we believe it is appropriate for university students to be at the forefront, sharing in the adventure of developing and applying new knowledge. The MPC79 designers not only had their projects implemented, but also had the satisfaction of being part of a larger experimental effort that will impact industry-wide procedures.

Looking Ahead

Many of the MPC79 student and faculty designers are already looking ahead, making plans for iterating their present designs, and dreaming up more ambitious projects. Many other students are eager to have access to VLSI implementation, so they too can have the experience of learning to design in a state-of-the-art technology by actually *doing it*. The demand for such services is building rapidly, and the accessibility of VLSI implementation is stimulating interesting new integrated-system research and educational activities in the universities.

There will be more MPCs in 1980 to service this university demand. "MPC580" and "MPC1280" are now being organized to service the university courses in the spring and fall. In parallel with these efforts, a transfer of the VLSI implementation system technology is underway from Xerox PARC to the Information Sciences Institute (ISI) at the University of Southern California. ISI will operate a VLSI implementation system and coordinate the maskmaking, wafer fabrication, and packaging for the universities in the future, with funding provided by DARPA (see also the *University Scene* column in this issue).

We see many interesting directions for future improvements in VLSI implementation services. Once a large user community exists, an implementation system could be operated as a continuous "server," "spooling" requests for implementation, and then, when a mask-set full of projects is queued, automatically merging the designs and initiating mask and fabrication runs. With sufficient demand, the interval between runs could be quite short, freeing designers from having to work towards a common design cutoff date. The system could handle other technologies such as CMOS and I^2L, if enough designers wanted access to those processes, and if standard interfaces between design and implementation were developed for them as has been done for nMOS. The service could eventually mediate a typical marketplace: those processes most commonly used by designers would have shorter queuing times and lower costs per implementation than would less common processes, but the less common processes would be accessible.

We are very pleased that MPC79 has provided a sufficient demonstration of the feasibility and practicality of remote-entry, fast-turnaround VLSI implementation, so as to lead to the funding and operation of a regular, scheduled VLSI implementation service for a substantial government-supported research community. We believe this service will provide a large return to the country on its investment by greatly leveraging the human resources to be applied to research explorations in integrated-system architecture and design. We also believe this service will provide an example that industrial firms can follow of a new way to "do electronics" that will enable rapid and economical development of many new commercial applications of VLSI systems.

Acknowledgements

We're grateful to all the folks who pulled together with us to make MPC79 happen, especially our friends at Defense ARPA, at Micro Mask, and at Hewlett-Packard/ICPL. We all owe a great deal to the dedication and efforts of the instructors and project coordinators in the universities, who, working under great pressure and on a tight schedule, did such a fantastic job with their design courses and project-labs this past fall.

We especially thank all the student designers for their enthusiastic response to the courses, and for their magnificent efforts and accomplishments on their integrated system design projects. Organizing and carrying out the implementation of the many imaginative designs in MPC79 has been a very exciting and rewarding experience for us here at Xerox PARC/SSL.

References

Clark, J.H. January 1980. "A VLSI Geometry Engine for Computer Graphics." Paper read at Conference on Advanced Research in Integrated Circuits, M.I.T.

Conway, L.A.; Bell, A.G.; and Newell, M.E. January 1980. "MPC79: The Demonstration-Operation of a Prototype Remote-Entry Fast-Turnaround, VLSI Implementation System." Paper read at Conference on Advanced Research in Integrated Circuits, M.I.T.

Conway, L.A.; Bell, A.G.; Newell M.E.; Lyon, R.F.; and Pasco, R.C. January 1980. "Implementation Documentation for the MPC79 Multi-University Multiproject Chip-Set." Xerox PARC/SSL Technical Memorandum.

Holloway, J.; Steele Jr. G.L.; Sussman, G.J.; and Bell, A.G. January 1980. "The SCHEME-79 Chip." Memo No. 599, Artifical Intelligence Laboratory, M.I.T.

Hon, R.W. and Sequin, C.H. March 1980. *A Guide to LSI Implementation*. 2nd Edition. Xerox PARC/SSL Technical Report SSL-79-7, Palo Alto California.

Kahn, R.E., ed. November 1978. "Special Issue on Packet Communications." In *Proceedings of the IEEE*, Vol. 66, No. 11.

Mead, C.A. and Conway. L.A. 1980. *Introduction to VLSI Systems*. Boston: Addison Wesley.