MOSIS -- THE ARPA SILICON BROKER[1]

Danny Cohen and George Lewicki

USC/Information Sciences Institute

This paper is actually an edited transcript of the talk presented at the conference. Many references to visual accompaniments, difficult to reproduce here, have been eliminated.

INTRODUCTION

The idea of a silicon broker was conceived by Carver Mead some time ago as a way to give a large community of chip designers access to fabrication services and as a way to speed up the fabrication process. MOSIS is the ARPA silicon broker that we have implemented at ISI. With MOSIS we are trying to totally isolate the designer from all the trivia that fabrication requires.

The main objective of ISI's VLSI project is to support the fast turnaround requirement of the ARPA VLSI community and of related programs. Another of our objectives is to help expand the VLSI design community by supporting research institutes and universities that are actively involved in VLSI. We hope to help MIT, Caltech, Berkeley and other universities train as many VLSI students as they can.

In addition, we'd like to encourage more vendors to offer custom VLSI services. We were pleasantly surprised at the number of organizations already in the business of offering those services.

For the time being, we are sorry to report we can serve only the ARPA VLSI community. However, other government-sponsored users may gain access to MOSIS by special arrangement with ARPA. If you are interested in our service and your project is government sponsored, please contact us or ARPA, and we will try to help you. Remember that NSF is part of the US Government, so people sponsored by NSF will probably be able to participate.

[1]This research is supported by the Defense Advanced Research Projects Agency under Contract Nos. MDA903 80 C 0523 and MDA903 81 C 0335. Views and conclusions contained in this paper are the authors' and should not be interpreted as representing the official opinion or policy of DARPA, the U.S. Government, or any person or agency connected with them.

MOSIS

The MOSIS system developed from an idea demonstrated in recent years by Caltech and by the MPC project at Xerox PARC. We'd like to acknowledge everyone who helped us, but this is only a partial list. We would like to mention Carver Mead--he was the first to put together many chips, and he taught us how to do it; and Lynn Conway, whom you have just heard telling us all about the MPC project at Xerox PARC. Thanks also to the fantastic crew there: Alan, Martin, Dick, Ted, and many others--it's impossible to name everyone here; please accept our apologies.

ISI's silicon broker works as follows: Users who have obtained access to MOSIS communicate directly with the system via electronic mail. Most users are on the other side of the ARPANET, whether across town or across the country. MOSIS understands various types of information, such as, "this is a description," "this is a pad," and "this is the technology we want," and it knows that CIF files describe the geometry of a project. MOSIS accepts several types of requests, for example, "please start a new project." All requests are very formal, because machines, not people, read them. Questions about sending requests to MOSIS can be sent to MOSIS@ISIF. The questions should be stated in plain English, e.g., "Please tell me what to do." The answers from us will probably be equally cryptic: "We couldn't understand your message, but if you want to talk to us, do such and such and we will send you the MOSIS User's Manual." Save time and trouble by reading the MOSIS User's Manual--it explains everything a user needs to know.

All user-provided information flows through MOSIS to MrBill, our geometry handler that checks CIF files, packs sets of projects onto a (smaller) set of dies, translates each die into MEBES format, makes bonding diagrams, and more. Ron Ayres wrote MrBill in ICL--beautiful language, beautiful system, works magic, very efficient. For example, it can plot CIF files like Figure 1. Figure 2 is a slightly more complicated plot. It's not clear exactly what MrBill drank before he plotted it, but we were told, it's OK, it's a bubble memory. MrBill's primary task is to produce tapes that the foundry uses to make masks.

After MrBill does his work, the next step in the process is mask fabrication. Mask houses expect two types of things from us: tapes with MEBES files and job decks. MEBES files contain the information that the mask house uses to make bitmaps (which are made into masks). A job deck, about one percent of the size of a MEBES file, maybe less, contains the specifications for each MEBES file--parity, record size, etc.

Fabrication itself is very simple because somebody else does it. Once the masks are made, all we have to do is drive three, four, or maybe ten miles in Silicon Valley with the masks to a wafer fabricator. (It is wise to drive slowly to make sure the masks don't break.) After that, if we're lucky--and typically we are--we end up with a couple of wafers.

Once we have the wafers, we like to probe each of the chips, not just all the wafers, so that no one will tell us later, "Maybe only the north part of

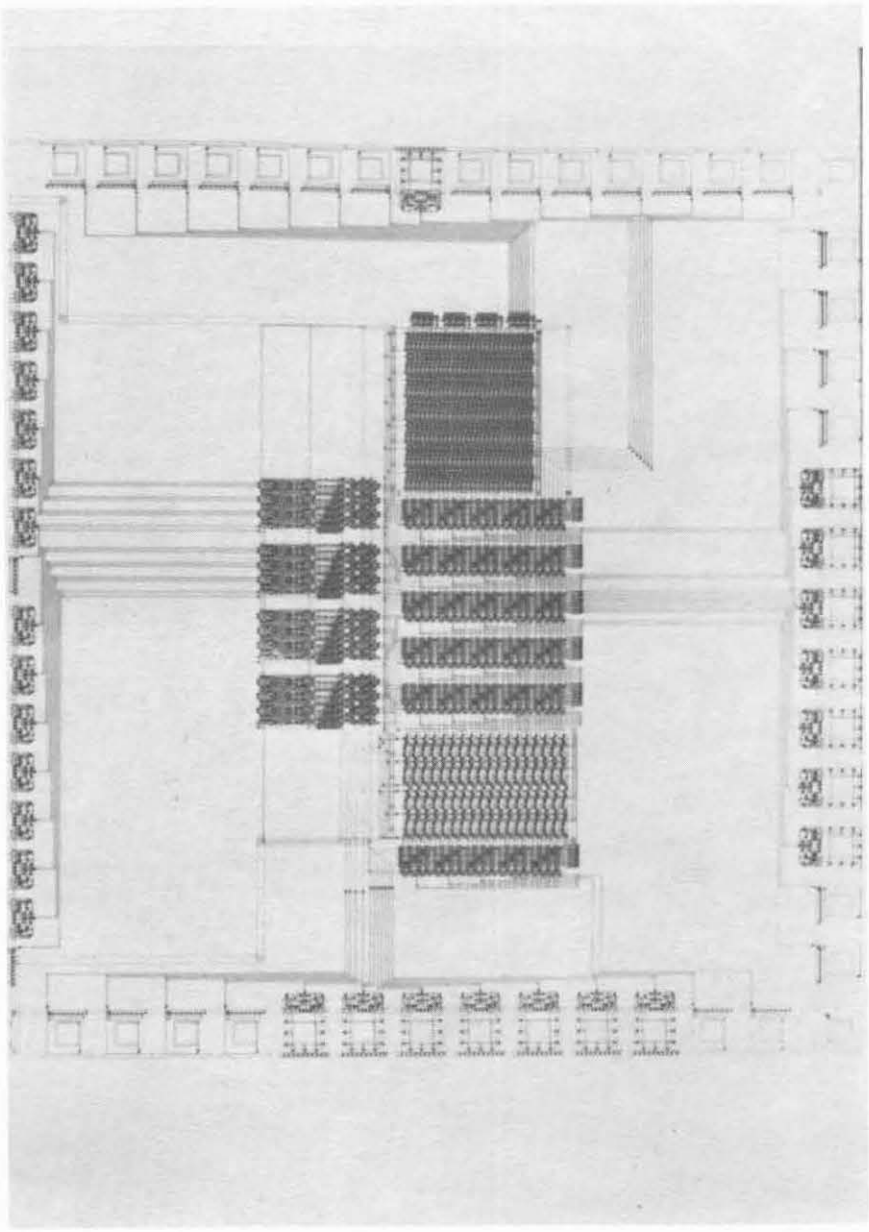

Figure 1: Simple plot produced by MrBill from a CIF file

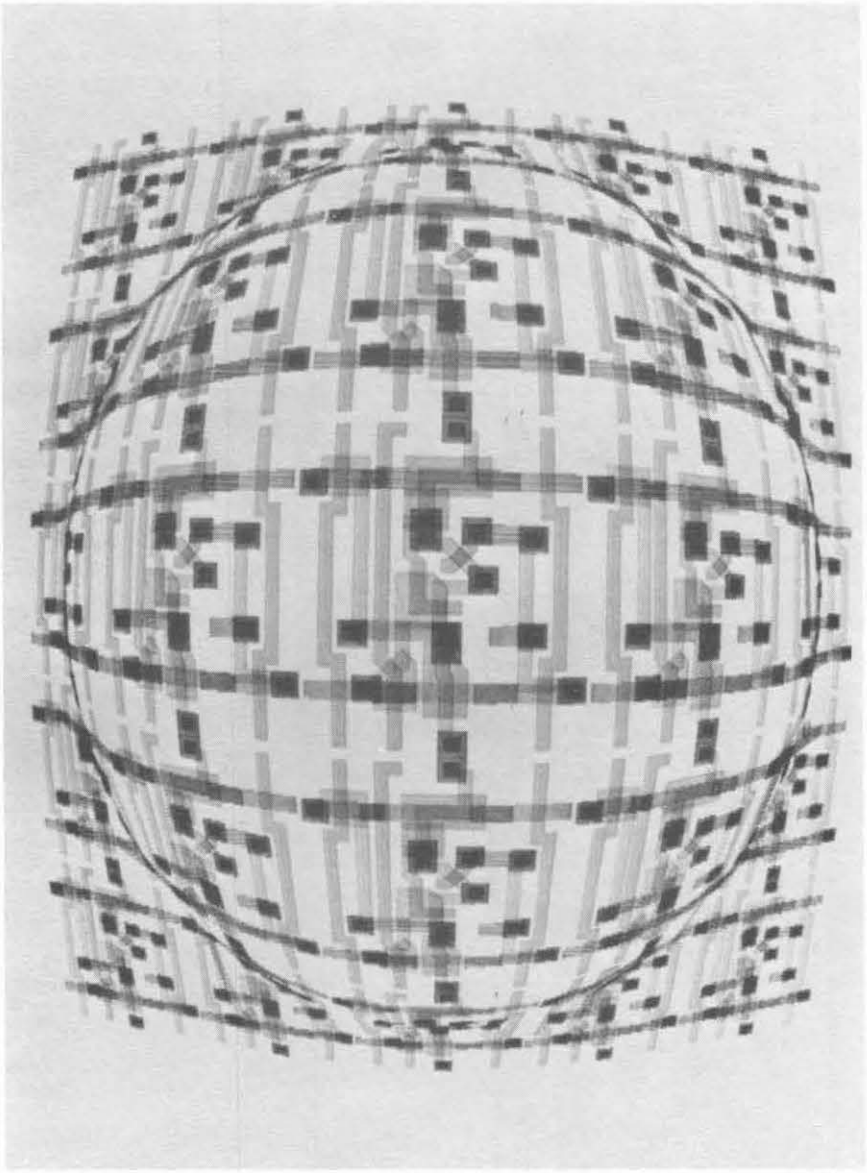

Figure 2: A more complicated plot of a bubble memory

the wafer is good; if you happen to have a southeast project on it, you probably lose." We like to make sure that the wafer is uniformly good.

Then we break the wafers into individual chips, package them, and run some more tests, if we can. Afterwards we distribute the chips to the users, and each user examines his chips, points fingers at everyone, resubmits, etc.

STANDARDS FOR MOSIS

We prefer continuous-spooling mode, which means that we don't like to advertise deadlines. Whenever we have enough projects, we fabricate them. The sooner you submit, the sooner your design will be fabricated.

For the time being, we support CIF 2.0+. If you know only about CIF 2.0, fine; we support it. There are several other features we support, and perhaps we'll eventually convince the entire community to use them.

At present, we support nMOS with the Mead-Conway design rules. More processes will eventually be offered. This means that we do not now support 2-layer metal, buried contacts, etc., but we will later. Currently we use lambda equals 2.5 microns. This is a feature size of five microns. We have talked to several people about smaller feature size, and we are in various stages of negotiations about smaller values.

Once given a file, we can change lambda a little bit, but not a lot. If a big change of lambda is necessary, the designer has to make the changes. If the design is for 2.5 and we have 2.0 available, we might change the size if the designer allows us to.

Our standard packages for bonding projects are 40 and 64. If we can find packages for 89 bond projects, we might be able to bond them, but it might take more time.

We try to provide fast turnaround by streamlining all the interfaces. We have been told by industry that if we pay a premium, we can get faster services. We are not sure this is what we want to do right now. With more money we know we can get faster service. We are trying to see how fast service is if we know about tape parity, registration marks, CDs, and all the other details that are required for fabrication.

PROBLEMS

We are constantly trying to improve the service from MOSIS. We like to try new software for its added features. We like to try new mask houses so we don't have to depend on one source. We like to try many fabrication lines. We like to change the way we test wafers, packaging, etc.

We have problems in the process of qualifying any changes, that is, making sure a change is really an advance. For example, the first problem we had was

deciding how to qualify a new set of software. The problem arises in comparing two masks, one produced by the old and the other produced by the new system. Are the patterns really the same? A microscope is supposed to help, but it can't do a good job. We tried many ways and finally worked out a very strange technique. Suppose you want to compare mask A and mask B. What we did was to overprint A and B bar [the reverse of B], and A bar and B. In this way we discovered all the changes. We did all the printing on one plate so we wouldn't have to use a special microscope.

We applied this test after we were sure there were no other problems. We were shocked by what we discovered--one little bug that turned out to be many bugs in the manuals, not a bug in the software. But it was just as bad, so we had to fix everything until we finally got all those squares to be exactly the way they were supposed be, blank.

We have also had problems in the preparation of a job deck. A job deck is a definition of a wafer. Figure 3 shows a wafer containing 18 different die-types. Each die-type requires six files, so over 100 files are involved in preparing this wafer. A lot of coordinating is therefore necessary, and we would like to make sure everything happens right.

Some of the companies we've worked with (Xerox, Boeing) share horror stories with us about the production of an accurate job deck. Our goal is to generate job decks with computers. Figure 4, for example, is an input for a program that generates a job deck (unfortunately, not for the wafer shown in Figure 3. Sorry about that.). The input contains the name of the run, like N11E, and the definition of each layer. The letters D and C determine the dark or clear mask--very important, or else you get some odd results. The name of the level and the name of the job have to be written correctly so the fabricators do not make mistakes. The input also contains some coordinate, and the map, which controls the position and the choice of over 100 files. With all of these variables, there is high potential for something to go awry.

We have to screen new fabricators carefully. Ideally we'd like to give them a form to fill in, and then we would continue from there, if their qualifications were close to what we expected. In reality we ask the fabricators, "What technology do you offer? nMOS? CMOS? What?"

We also ask, "What are your design rules?" Actually, we don't ask, "What are your design rules?" We say, "Those are OUR design rules. Do you support them?" And sometimes we get answers, "Yeah, we support them. Lambda equals, say, two microns, for everything except ..." and we say, "Too bad. You don't really support our design rules for this feature size...." We have to decide at what feature size our design rules are supported.

Next we ask the potential fabricator for electrical parameters, everything we need to know about masks, polarity, bloating, etc. Then we tell the fabricator how we like to measure. As a matter of fact, selecting a fabricator is not quality control--it should be a process control, insurance that everything meets our standards, including turnaround time, and, obviously, the expense.

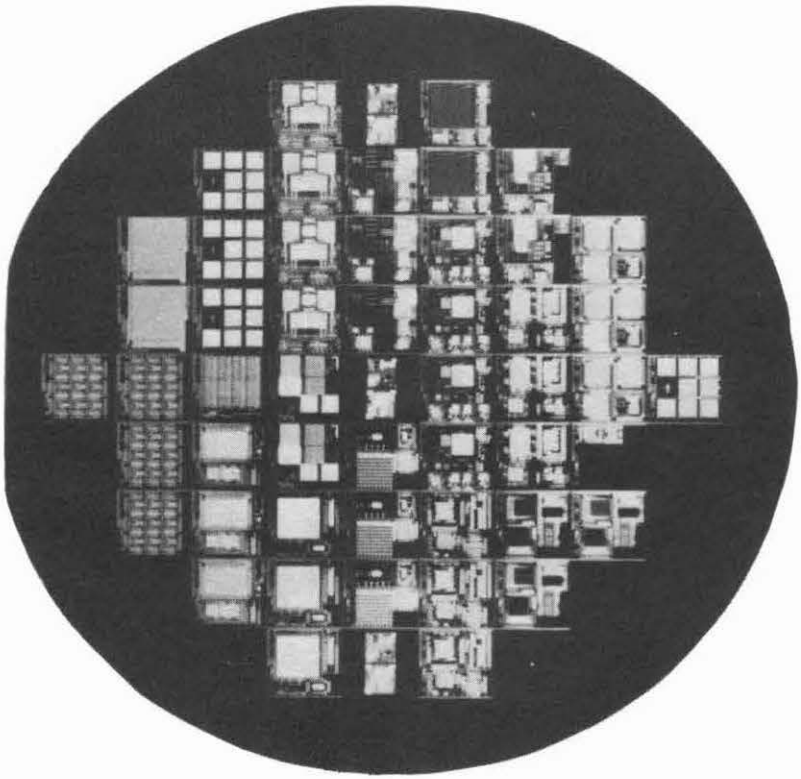

Figure 3: Wafer containing 18 different die-types

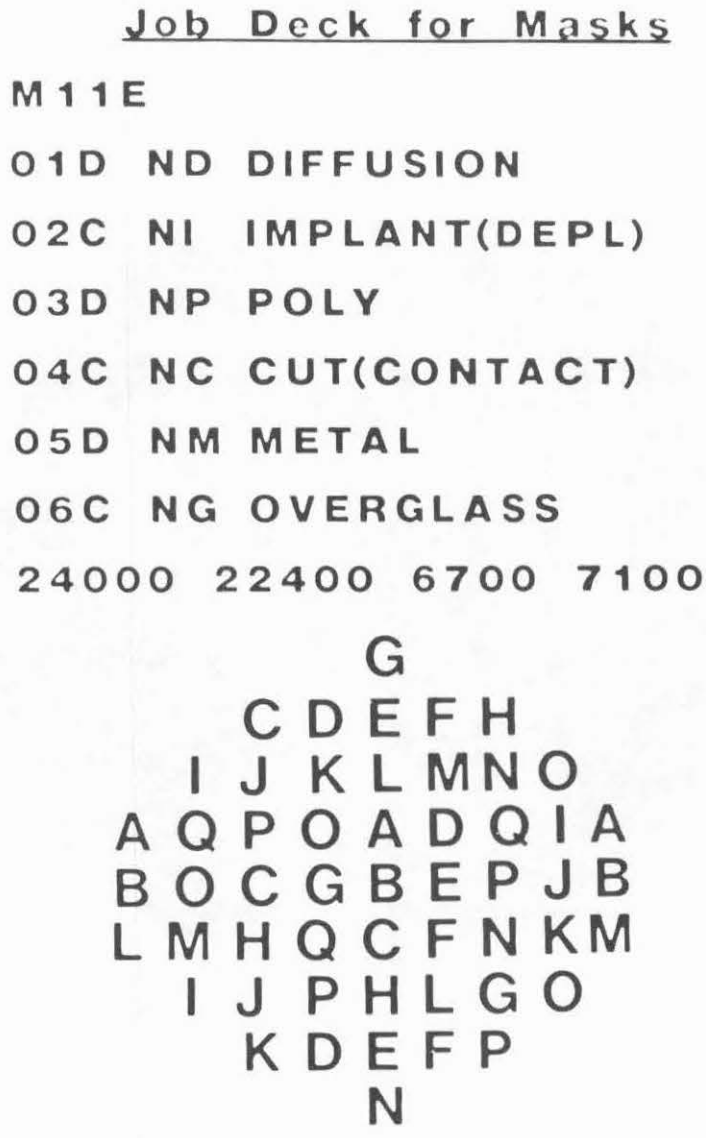

Figure 4: Example input for a program that generates a job deck

We specify five electrical parameters to the fabricators . We ask that the enhancement devices' threshold voltage be about +0.8V. For depletion, it should be about -3.5V. For k=4 and for k=8, minimum size inverters Vinv should be about +2.0V, and the poly sheet resistance should be less than 50 ohm/square.

Obviously, this process control doesn't cover everything. Our expectations are that, when we go to reputable lines, they know how to do the things we have to have done in order to support our designers and they will produce good chips all the time. We would not be surprised if one day we find wafers that don't do anything right but that do meet these five criteria. When this happens, we are not going to sue anyone--just say, "Sorry! We are not going to use this line anymore." We are not interested in finding out exactly why something went wrong.

TESTING

Next, we have the issue of testing. We don't need tests that are designed to calibrate fabrication lines because we don't care to calibrate fabrication lines. There are already people whose job it is to calibrate the lines. The tests that are important to us tell us something about what yield or what performance users can expect with our design rules; those areas are our concern. We would love to have standard industry wafer-acceptance procedures. We'd like to be able to design one test chip into every wafer, accept the produced wafer, and then test it. If it passes the test, we say it's a good wafer, and we pay. If it does not, we don't take that wafer. We'd like to establish a standard that will be accepted both by industry and, obviously, by users.

Unfortunately, we have not reached that point yet. We are working on a standard. JPL, Xerox PARC, NBS, and the Integrated Circuit Lab of HP are participating in this effort. We have made some progress, but, again, we are not there yet.

First of all we are trying to test the basic elements, like transistors. Then we like to test the building blocks, like inverters, to see if they work to our specs, and then even more complicated random fault structures. In order to do that, we have our "standard" test patterns, which are designed for probing, not for bonding.

That test vehicle is an interesting camel. It was supposed to be a lion. The committee that designed it met too many times. It's very complicated. It has had many tests, and we are trying to simplify it. Maybe we will be able to turn it into a lion again. But we don't know yet. We will work on it.

Another issue is how to verify the completeness of the testing. We would be uncomfortable in a situation where all the tests are passed with flying colors, and no device works, or most devices don't work.

In the past, there have been two situations in which tests were perfect but devices didn't work. When that happens you say, "Gee, that's too bad. Let's change the test." We don't want to go into too many details, but let us describe one of those cases to you.

On a certain run we had, among others, die D and die E. Unfortunately, an error crept into the job deck when someone at the mask shop decided to retype it manually. When the person retyped it he interchanged the diffusion level of the two dies.

However, our test patterns on the two dies are identical! When we probed, everything was perfect! So now we say, "Aha! If we put the die designation on all layers..." (by the way, we thought about doing that before but never did or the error would have been caught). Now we know how to find this error. But we have no way of knowing how many other problems will pass all our tests without being caught.

We use both small test patterns that are part of every die and a few bigger drop-in tests. All the test patterns of all dies on all wafers are probed. We actually probe every die. We compute the mean and standard deviations over the sample of 46 to 50 dies. We are looking for some interesting patterns, but we hope never to find them.

Two- and three-dimensional analyses of problem data do not reveal any significant inter- or intra-wafer pattern. Two-dimensional analysis, for example, indicates whether the north part of the wafer is better than the south or whether the middle is much better than the edge. Three-dimensional analysis shows up a difference between wafers. Maybe one wafer is OK, and other wafers are not. We compute both by wafer and by position, and by many other statistical means.

We have been very delighted not to find significant patterns. If we found significant patterns, for example, that the northeast corner is always the best, we would be flooded with requests from users: "Please put my job on the northeast," or "put mine on wafer number three," or something like that. We believe that Monday wafers are not really as good as Tuesday wafers, but we cannot prove it!

We have also experimented with several comprehensive structures that test typical user devices. Years ago, there was a notion of a typical picture for computer graphics. A thousand lines was considered a typical picture, and everyone was supposed to support such a typical picture. What we need now is a typical user device that we can put on every wafer, try it, and if it works, then everything is OK. If it doesn't work, we have a problem. We are still looking for such a device.

As a matter of fact, we are designing several canaries[2] just for this purpose. One of our canaries is a 19-stage ring oscillator that Xerox's MPC used. We'd like to use it as many times as we can. One of the neatest things about this ring oscillator is that it uses only three pins. This is very important, because we can nearly always bond it in addition to other projects that don't require all these pins. When we bond projects we always try to bond the 19-stage ring oscillator and to test all of the oscillators to see that everything works.

In addition, we are trying to get some yield information from the world's slowest 4K RAM. We try to come up with ratios such as 3 bits out of 4K didn't work on so many units to see if we can derive some yield statistics that are meaningful for users. Anyone who has entries or suggestions for this collection of canaries is most welcome to submit them for consideration.

Thanks are due JPL and NBS for providing the following test structures. Figure 5 shows die F of run B; what is evident here are some random fault structures. There are several miles of metal over poly, etc. One of the interesting things to see here is that this is both a drop-in as well as a user device (at the lower right-hand corner). We don't really make any differentiation between user projects and drop-ins.

There were random fault structure dies with more miles of step coverage, and their logarithmic connections were visible. With random fault structures like this, there is always the hope that the small portions of the structure are small enough not to have any faults and the big portions are large enough to make it easy to find the faults. And the worst that can happen is that all of them fail or none of them fails. Then you know you are looking at the wrong range.

We had another interesting drop-in from NBS. It is interesting because several test patterns are repeated with variable geometry. It is revealing to learn more about the geometry and compare it with claims made by manufacturers.

PACKAGING

Our standard packages, as mentioned earlier, are 40 and 64 pins. We might bond several projects in the same package if they go to the same customer. It often happens that several small user projects can be bonded together. We always try to bond as many test structures as we can; for the time being this includes only the 19-stage ring oscillator. If we have more later, we will try to bond them too, but never at the expense of the paying passengers.

[2]Canaries used to be employed in underground mines as indicators of air quality. If the air was bad, the canary would die, but the miner would have a chance to return to good air.

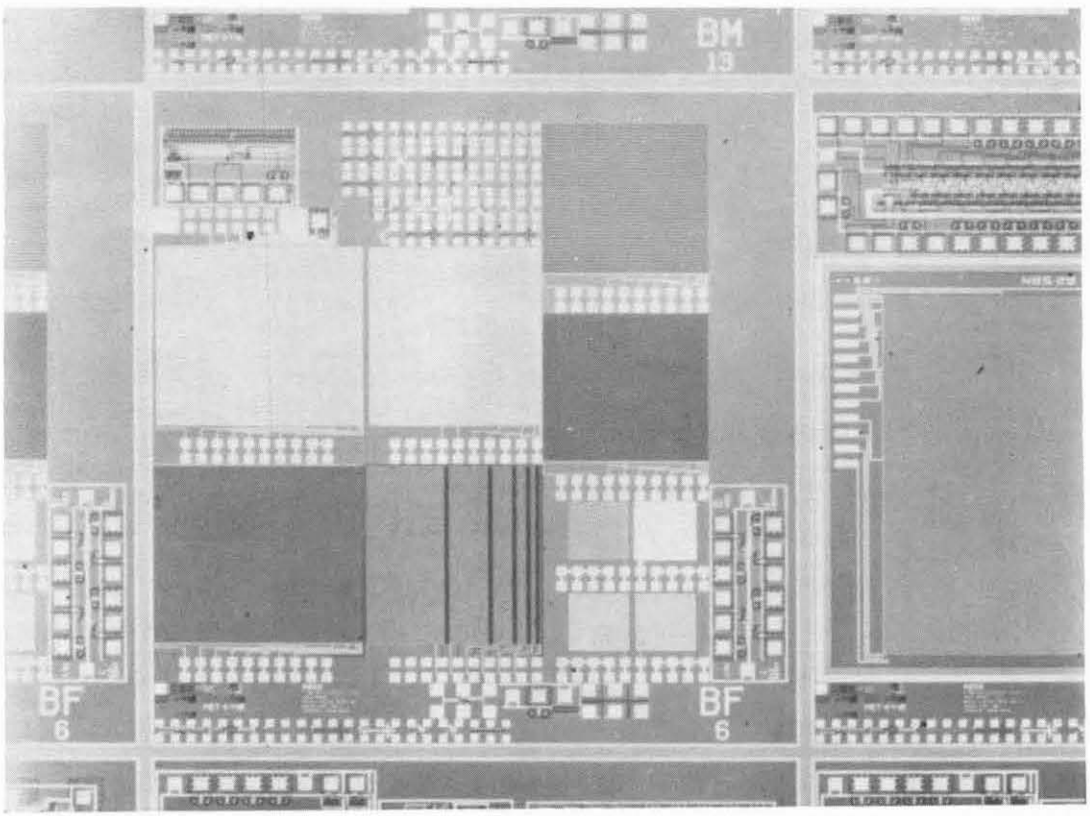

Figure 5: Die F of run B

Researchers at Lincoln are pursuing several techniques for bonding standardization without imposing limitations on the designers. At present we bond manually. We'd like to be able to go to automatic bonding, but the large variation between projects prevents that, at least at reasonable cost. The folks at Lincoln are trying to work out some techniques that will enable us to bond automatically.

Our strategy for die distribution is as follows. We try to fabricate n copies[3] for each project, such that the probability of giving a designer at least two dies that are mask defectless and silicon defectless will be greater than 90 percent. Now maybe 90 is not a high-enough number, but 90 will have to suffice. In the worst case we can always refabricate the die just a few weeks later. The die should also be bonding defectless. If the bonding is not 100 percent, there is no point in having a perfect project. From time to time we have discovered some problems in bonding.

Using the available data, we were able to achieve our goal of 65 projects on 18 die types with one wafer set. We showed this wafer set before (Figure 3). This die J (Figure 6) happens to have eight different projects on it. Some of them could be bonded in the same package (utilizing unused pins), some of them not; the arithmetic of how many of each can become very interesting.

One of the most interesting projects was done by a student of Chuck Seitz, Eric Barton. Eric was very impressed by Chapter 7 in Mead and Conway, and he decided to do a self-timed project. So he had his own clock wired into the project; it can be seen in Figure 7. When power is applied, the hands actually move. We disconnected it at 8:00 this morning so it still registers that time. The clock was a great thing. We never knew about it until we looked through the microscope. First time we saw it under a microscope we checked--it was three minutes slow.

CONCLUSION

We want to push lambda but not at the expense of the design rules. We would like to see if we can really reduce the design rules. It's nice to say one micron, but obviously we do not have to stop there. When people from industry come to us with submicron processes, we will be delighted to check each of the processes. We're always ready to add more features to nMOS and always willing to use other technology. Though we're not sure in exactly which order...

By the end of 1981 we expect to support over a thousand designers and a thousand projects. We want to be very careful with this kind of prediction. We like to think that we are underestimating: we would like to see more users. Please, feel free to try us; we hope that we will be able to accommodate most of you.

---

[3]Needless to say, this magic number n depends on the active area of the project.

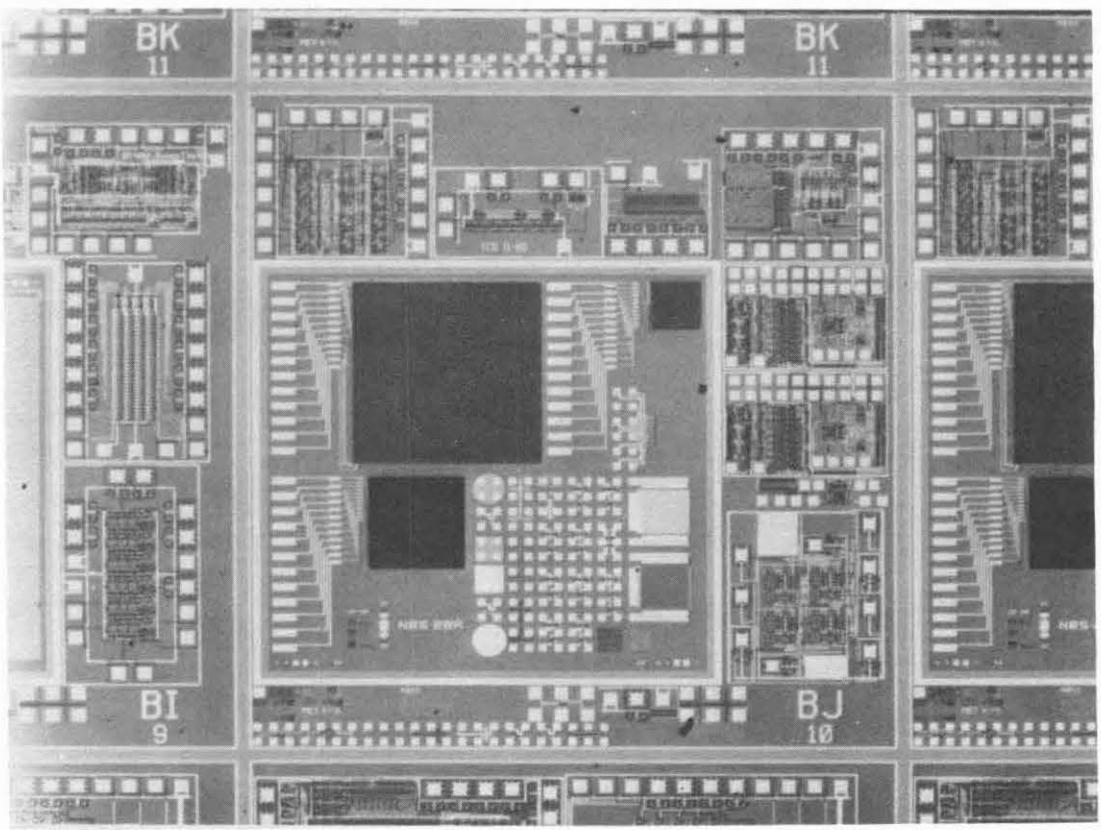

Figure 6: Die J with 8 different projects

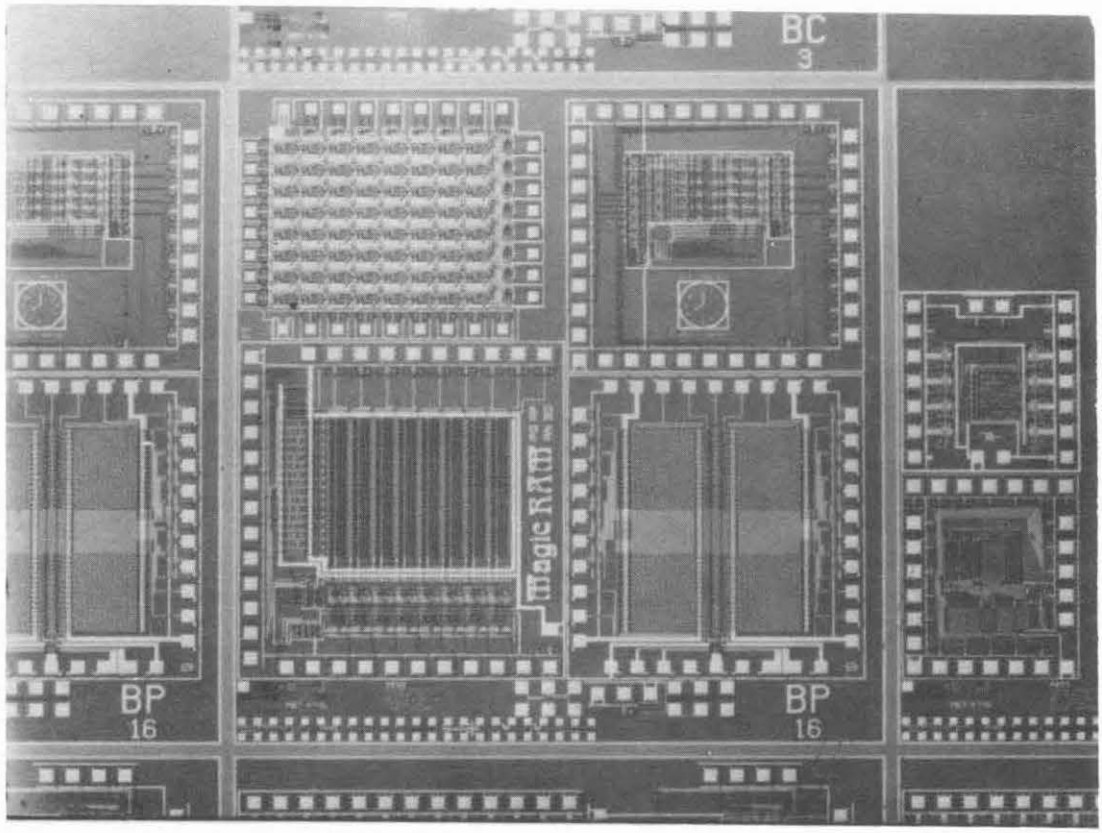

Figure 7: A self-timed project

ACKNOWLEDGMENTS

Thanks to ARPA, which provides all the funds for this project; JPL and NASA, which helped us a lot in the first run; NBS, PARC, and the HP Integrated Circuit Lab, which has helped us a lot in the testing. Thanks also to the crew at Caltech who provided the transcription of the talk and to Jim Melancon at ISI for editing the transcript into a reasonable form for publication.