Dr. W.R. (Bill) Bottoms

3MTS.COM

I was told that I should speak for a certain amount of time and then there would be time for questions, and it would be a lot more fun if you would just ask the questions as they come up. That way, I am more or less assured that you do not finish before I do.

So the area that I am going to focus on is to, first, remind us a little bit of what we already know. So those of you that are still eating, you should not stop right now because you will know most of what I am going to talk about in the beginning, but the issue in front of us here is that test packaging, and I should include burn-in costs, are now the limiting factors in the cost of integrated circuits for many device types; pretty soon, it might be all device types, but it is approaching that today. And the issue is what can be done to reduce costs. If there is a motivation to use test-in-tray, it has to be because it will allow us to do what we have to do, which is done anyway, but it will allow us to do so at a lower cost. Just to remind you that, over the 40 years from 1967 when CMOS began to be used, until today, well, actually until 2007, we saw one million times improvement in the density of integrated circuits. We saw a million times improvement in the cost to build an integrated circuit, and we saw a million times improvement in the feature size.

Now, it probably goes without saying, but there has been no activity in the history of mankind that has ever achieved that kind of progress in a 40-year period. And the only things that even came close are things that were enabled by what has been done with the semiconductor industry. So we sat under that laurel for just a little while, and then we

decided what we were going to do for the next 40 years. But you might think about it for a moment, what happens if we have another one million times improvement over the next 40 years? And,so far, people don't seem to be backing off of that. The world will be a very different place.

Focusing on cost, the cost of a transistor is reduced by a factor of a million, really kind of hard to believe, and during that period it has caused many things to change. For 40 years, we knew exactly what the changes were going to be: we are going to shrink CMOS, and that increase in parallelism that comes from the shrinkage and other things we do are what enables the cost reduction to take place. So there hasn't been a mystery about what will be done. Occasionally, there have been little short periods of uncertainty while we figured out which solution would be used, but we basically knew where we were going. We're going to shrink geometries, we're going to improve the designs for better efficiency, increase the wafer size, and that also increases parallelism, and the focus was clearly on design and fab. And the reason the focus was on design and wafer fab is that test and burn-in really didn't amount to very much in terms of the total cost of delivering the product. And so far, in most of the history of this industry, we have been in that situation where people weren't paying attention to test and burn-in costs.

The fundamentals are now changing and perhaps I should say they have already changed in most areas. And the things that we knew to be fact are no longer true. So the first fact that we knew is that CMOS die shrink is the path to cost reduction. And in order to maintain the pace of scaling -- Moore's Law -- you have all seen the "More than Moore" diagrams of various types that come out of the ITRS, and I do not really run mine past 22 nanometers. I realize that people have roadmaps that go down to 9 nanometers or

something like that, but I wouldn't hold my breath to see any of those make it to market. There are all kinds of reasons why it doesn't make sense, and perhaps the largest reason is it will cost more to push that next node, than the benefit of achieving that node in terms of cost reduction to be a cost increase.

So what happened is well known, functional diversification was added and therefore we could continue scaling, in a sense, by using functional diversification to put more things in the same package, and therefore get an increase in functional density without having to pay too much for it. And so the model says that we continue Moore's Law scaling along that path. At the same time, there is a third vector where we have SOC and system-in-package that allows us to bring the interface, the human being, a little closer into the package, and that further reduces the size and cost of the systems, the power required, all of the elements that have been driving progress. But, you know, that still is not enough. We have to use the third dimension if we are going to maintain the kind of progress in cost per function and functional density, and it turns out that people have been using the third dimension for quite some time. These were not all made at the same time, but you can see what these are. This is a sequence of wire bond solutions. And I think the ultimate wire bond solution that has been published is the 24-layer memory product from Hynex, and I only call it a product because it actually said that in a press release, but I don't think they have built but one, and right now I wouldn't hold my breath for them to build another one; while you could build it and it did work, it isn't economically viable; you have gone past the point of no advantage by going further. So we have to find other ways to access the third dimension if we're going to maintain CMOS, give up on the pace of two-dimensional scaling, and use the third dimension to increase the density of the devices that we are manufacturing.

So there are several examples of that here. This just shows how these are stitched together when we require the same wire to go to multiple places. This (illustration) is TSV with Ziptronix, which shows two layers that are stacked face-to-face. This is a TSV which has stack-up of layers; you probably are all familiar with these companies and know what they're doing. There are so many different versions of this one, and somehow I keep pulling out the Samsung one, but all of the large companies now publish their version of this; this is 9 silicon die interconnect with TSV. And two weeks ago, I think it was, maybe it's three weeks now, Elpida announced a product which is an 8-layer device for memory, that is packaged in a package that has the same footprint and the same volume, X, Y and Z, of a single die. And how do you do that? Well, you do it by making the die really thin and stacking them together and interconnecting them with TSV. And I was actually given a lecture at a meeting where they came and announced this, and during the panel in the evening there were several people that said, "Well, you know, they're not going to use that for production, that's just a demonstration thing." Those of you that know Elpida, if it is just a demonstration thing, they probably do it in Sagamihara [Japan], but if it is a production thing, it is done in Hiroshima, and this is all done in Hiroshima. I talked to the person who actually came there and announced it, and he says that it is tool for volume and it is going to volume. So now we're really going to start seeing some of the things that people have been talking about for a while hit the mainstream. How long that takes, I wouldn't venture a guess because they haven't announced it yet.

(Question): Yes? Please. Elpida's memory is primarily specialty DRAM, so they build a lot of different codes because their products are focused on areas that Samsung isn't focused on. And so they build -- if you've got DRAM in a mobile product, you've got a

really good chance that it's going to be Elpida, and if you've got DRAM in something that's more stable, or even the laptops, then there's a really good chance it's going to be somebody else.

So I'd like to present now another project that's underway. This project was defined to try to identify the roadblocks from hitting really high density things that, even though I've been working on this for a long time, it still feels like a science fiction thing, so this is Terascale computing. So far, the only thing you can see is a substrate. The "C" and "T" (graphic) stand for Conducting Nanotubes, and the reason you should keep that in mind is that, for many applications, copper nanotubes are a lot better than carbon nanotubes. And so we should be Agnostic -- it is just a conducting nanotube. And we use that to make sure we get a thermal short, because this is going to make a lot of heat. And then we stack something on top. And if I had more time, I'd put these layers on one at a time, but this is a processor with 1,000 cores. It's cut into 10 layers so that the interconnect length is really short. The frequency is actually pretty low, it is 1 GHz, so that we can deliver a teraflop with a power supply voltage of 400 milabels*. You will see some other things in there and I'll go through and try to explain them. If you're really going to make this a general purpose computing system, you have to have, according to Gene Amdahl*, one byte per flop, so if it's a teraflop, then you have to have a terabyte. That means we need 18 terabits of bandwidth for memory if we're going to feed it without any limitation in data IO. And I would point out something that you probably already know, but almost all the high performance computers in the world today all have limitations in the bandwidth density of IO. So the next thing is a little piece of memory on the top, this is 200 gigabytes of memory at 400 gigabytes per second, and it is divided into 16 sectors. Then, there's a memory cube sitting on the other

side, which gives you a lot of memory in the same space. This one has a little bit more than the Elpita part, but not much. If you look at this closely, it's 10, Elpita is 8, and therefore it is still about the same thickness of a regular die. I couldn't put any labels on it and make the thickness of a regular die and make it look like it really would, so we had to interconnect this some way, and it's interconnected with TSV's. And TSV's have wonderful characteristics, and they've got offsetting big problems. One of the problems with TSV is what happens if I've got a copper TSV going through a semiconductor layer, and it's close to a transistor that has stress engineering as part of its behavior. They don't change its temperature. So how far away do I have to be before my stress engineering gets safe? How close can I be before the stress engineering is blown up? And today, we don't know. But it appears that we actually get more freedom from those problems when we make the silicon really thin, and if any of you know the answer to that, I can tell you that we have indication that we don't really know the answer, but the indication looks like it gets better when the silicon gets really thin. So we still have some things to do.

There is something I didn't say anything about; there are interposers in there. There's no way you can use this without the interposers, several reasons why: First of all, if I'm running even a Gigahertz clock and I've got a thousand cores, plus memory to feed them, I don't know how many of those transistors are going to switch at one time, but I surely don't want to provide the power integrity limited by the conductivity of those TSVs. And so every few layers, you have to put in a large capacitance and maybe passive network that does other things, as well, so that you don't have any power sag -- let me remind you, it's 400 millivolts, you can't have any power sag at 400 mV before the system loses its mind, so you have to have them for that purpose. But it's not the main reason they're in there. The main reason

you have to have them is that, if you don't have something to take the heat out at intermediate levels, the silicon will melt, and the devices just don't behave very well under that kind of stress. So these interposers are actually microfluidic devices that have an alcohol pumped through them at sufficient pressure, and the vaporization temperature -- you can choose an alcohol to vaporize at any temperature you want, given the right pressure, so that they vaporize just below the maximum junction temperature, so you have very good heat conduction out, but it doesn't take much space.

So somehow now, if this is going to be a computer, we have to communicate to the rest of the world, so there's a little compound semiconductor device over here; it may or may not actually make photons, maybe it is only modulation-demodulation, but it does the conversion -- actually, to be more specific, in my mind, it doesn't make photons, period, because you can make them somewhere else and modulate them, but it does have a wavelength multiplexed, single-mode fiber, that gives it plenty of bandwidth, and it does make the conversion from electrical to optical so that, once you get past the length where it's energy-cheaper to move to optical, you can do it. There have been a lot of people that have published about what is the distance of transport at which it becomes power advantageous to convert to optical. Does anybody know the answer to what that is? Well, all the experts know the answer, but depending upon which expert you ask, the answer is different. So I've asked a lot of them, and the shortest number you'll get from anybody is 6 mm, but the longest distance you'll get from anybody is 3 centimeters. So if we're going to have the lowest energy, the lowest power requirement, then we've got to go off-package with optics. And in order to keep the thermal load in this little package as small as possible, then we don't generate the photons there because that takes a little thermal load, we'll just do modulation-

demodulation. And then we route through this substrate to get to the memory and connect the memory to the processor. So everything in there is something we can do. Everything in there has a precursor that is existence proof. But nobody has every integrated it. And even our interposers, those have been built, but just one layer, they haven't been laminated, they don't have any TSV's running through them at the present time. What this does is point out what the limits are, and I can tell you that, when this project started, the energy required to run this little computer was 20 times higher than it is today. When it started out, it was only 100 cores, the voltage was 600 mV, the distances were much greater, and so several things changed. We shortened the interconnect distance, we increased the number of cords so they operate at lower frequency, it was 10 GHz, now it is 1 GHz, and it is still a teraflop, but it takes $1/20^{th}$ of the amount of energy it took in the first simulations. So, just the final end story to that, obviously testing is a problem -- how do you test all those pieces? If they aren't known, good die, you've got a problem. And I'll say a little bit about that because maybe having known good die when each one of those layers may have a billion transistors could be a hard thing to do. But the limiting factor in digital systems is not frequency; the limiting factor is bandwidth density for IO*, and power, everything else is solved -- or maybe I should put it differently -- we hit the wall on those two way before we hit the wall on any of the other things if we keep going without boundary. So we've got a 2 terabyte optical transceiver in this compound semiconductor. So this is a teraflop computer, it's got enough bandwidth to be general purpose, Intel has agreed with that, IBM has agreed with that, HP has agreed with that, and a bunch of other companies that are part of a consortium that's been working on it, and nobody really agrees with all the details. It's a mix of things, but they kind of signed up to it. The idea is this could be built by 2015, whether it is built is a totally

different question, but it brings out the problems and it brings out a lot of the problems that can be addressed in some way by test-in-tray, perhaps. And I'll say a bit about that at the end.

(Question): Yes? We looked at all the one-dimensional conductors. The problem is the space required is a little harder to come by with the density and adding all the TSVs in there and the decision was that we'd use micro- fluidics. There's a lot of heat generated. The only thing this doesn't show is where the heat goes, that plumbing is left out. All those easy things are shown; the hard stuff is left out. But, yes, we have looked at all the -- everything that I know, and everything that group knows, about the best thermal conductors, and how they can be wired. But we need to take the heat quite a distance away. So if you have some ideas, we can talk about it offline, and I'd appreciate the help.

I put this chart up, which comes from the roadmap, and it's dealing just with wafer-thinning. And I didn't mention it, but in the previous slide, all of those die were 25 microns thick -- they could be a little smaller, but that's what we chose. And by the way, that whole computer, any idea how big it is? Including its packaging and all the other pieces? It's 5 cm X 3 cm X .5 cm. So it fits in your jacket pocket comfortably, if you don't mind the heat. [Laughter]

So we thought wafer-thinning was going to be hard. When we first put it in the roadmap in 2005, we had about five years before it was going to be broadly adopted. When we got to 2007, for the next re-write, everybody was already doing it, and they could do it on 300 mm wafers and you could read a newspaper with uniform lumens across the whole 300 mm, so it was working fine; it turns out, it's real easy.

Yes? 400 millivolts, pardon me. Yeah. I don't have a good number right

now, it's well within the capability of the micro fluidics, they have got a lot of capacity, but I don't have a good number. The reason that it's 400 millivolts is that, once you slow it down to a GHz, with these small geometries, this is modeled on a 22 nanometer -- true 22 nanometer process, now everybody is playing node games, so it's hard to know what a number means, but this is a true 22 nanometers. And the assumptions that were incorporated into that were the 22 nanometer assumptions and by dropping the frequency to 1 GHz. We can now take the voltage down to pretty close to the sub-voltage threshold because we don't care about frequency, it's only a gigahertz. And for 22 nanometers, it is almost hard to slow something down to a GHz. So it's a compromise of all those things together that allows us to drop that power by a factor of 20.

So even though thinning wafers is easy, handling them is not. After you thin them, now you'd got to handle this thing and if it has metal on it and you let go of it, it's going to curl up. And I don't mean just wrinkle a little bit, it's really going to curl up -- maybe at 50, it wrinkles a little bit, 50 microns. At 25, it curls up. It's at least a serious potato chip, but if we take it down to 10 or so, you can wrap it around a pencil and it'll stay there. And I suppose that if we put half the metal on one side and half the metal on the other side, we could make it flatten out, but that's an increase in processing cost. So we've got to find out a way to handle thin wafers and thin die. And, once again, maybe test-in-tray is a promise to let us address that.

So the cost -- that CMOS die shrink is the path to cost reduction, that's not true, it used to be. Where we are today is that, within the next decade, we'll find the cost of shrinking will exceed the benefit associated with the shrinking of CMOS. And so shrinking is over within the next decade for CMOS, and maybe less than that.

The next one is transistors don't wear out. From the time that we began the transition from vacuum tubes to transistors, we went from a world where the tubes wore out and the transistors never wore out, so that was an axiom in our world. But not true anymore. We've got some issues still at 45-nanometer. 45-nanometer devices wore out, but they have a long lifetime. And if they're designed correctly -- one point I was going to present the transistors and all the pieces, but it takes a little longer, so I left it out, but one of the major impacts in these changes, particularly as we go below 45-nanometer, is that what we've relied upon all this time is no longer true, transistors below 45-nanometer will wear out in what would be defined as a normal use lifetime of these devices. And everybody is doing things to their particular devices to try to reduce the rate of wear-out; obviously, driving the frequency and driving thermal density reduces the rate of wear-out, decreasing the current flow reduces the rate of wear-out because of less problems with electro-migration, but they're going to wear out. So "transistors don't wear out;" that one is gone. The lifetime limitations exist today and, with each note* [node?], they're going to get more and more stringent limitations on lifetime, as long as we are continuing to run a CMOS structure --

(Question): Yes? You know, I wish I could give you a good answer. Let me tell you what the problem is. The transistors that are under the most stress are the logic transistors. And the logic transistors in a multi-core device like that, we have no idea which one is going to get the load. And so, if they switch every cycle, every transistor, they're going to wear out reasonably fast. I mean, it's not six weeks, but it's less than 20 years. But they don't switch every cycle. And the way the controls are set up for that, I mean, I even hate to say this because it scares me to death, that you couldn't possibly do it, but the people that do it say it works fine and that is you keep a number of spare cores and their power is

completely turned off, and there's a transistor in every core that reports back its temperature; when it begins to get close to the maximum die temperature that the transistor is designed for, the junction temperature, then it will light up a core that's standing by, it will double the memory and route it to both cores, and run through the instruction depth, and then turn off the hot one and the cold one takes over. And so, people have tried really hard to structure something where they could predict which core is going to be the load so that you don't have to give each one of them the same kind of thermal management capability, and I think everybody has pretty much given up on that. If it isn't general purpose, then it becomes too complex. That's not based on anything I did, I was told that by an expert.

So the next thing is power requirements for transistor-based electronics are really not critical issues. And it's easy to see how we got into that assumption because, when we went from tube to transistors, we gained a few orders of magnitude of power, and that left us in good shape, and then we just saw how the power density went down by a factor of a million, so it can't be that power is a problem, right? But, of course, you know the answer to that. Portable consumer products live on batteries. If you could make your battery last a little longer than your competitor's battery, you're going to get the business, and so there's a big pressure on battery life, and everybody buys the same batteries, so the battery life pressure is to reduce the energy usage.

Videogame consoles, you know, the most advanced microprocessor for at least two or three years was a cell processor, made for a videogame. And the biggest failure rate they had with the cell processor, you know what it is now because you are reading this slide, you might not have known it before, people would get these game consoles and they'd put them in their bookcase between all their books and shove them back against the wall.

There's no thermal exhaust, and so -- and they're really hot chips -- and they couldn't sustain the temperatures, so they'd get thermal shutdowns all the time. I had mentioned this in a meeting where there was somebody that asked a lot of questions, and they seemed -- some of the questions seemed almost a little belligerent, but I answered them all as best I could. And then we broke for lunch, and I was giving a keynote address, and he came over to sit down with me and he said, "You just trashed my product." And I said, "Well, wait a minute, was there anything that I said that was wrong?" And he thought about it for a moment and he said, "No," I was right. And I said his product was the most advanced microprocessor in the world, but obviously had a problem for its application.

Everybody has read about server farms. We can't put them just anywhere because the power required to drive them exceeds what's available on the grid unless you make special arrangements, and so anything that can reduce the power draw and the server farm has got big value. And so we now have to say that the cost of power can no longer be ignored and it's now one of the factors that used to be okay in transistor power, not a big deal, and now even for logic in memory, it's a big deal. So all these things have eventually been crossed off. So that's gotten rid of the history, is all the things we thought we knew, they don't work anymore. It's also true, if you remember in the earlier slide, I said that the cost of test in assembly was not a significant portion of the cost, and that's no longer true; test in assembly now account for, in some cases, a majority of the cost.

So this is an SI chart from a couple of years back and work has been done to change it, but it was useful to use it anyway because it shows what I want to show, and that is this is the cost of fabbing a transistor over several orders of magnitude, but that's only from 1980 to 2010; and this is the cost of test equipment depreciation to test that device. And no

matter what you do about which unit to use on which one, there is an inescapable fact, and that inescapable fact is that, over most of this period, the test cost has not scaled. The cost of building transistors went down, the cost of confirming that you built them correctly is going up, it is not that it's not going down as fast, it's going up. So what's been done to rest this rising cost of tests? And, actually, people have been working on it for a long time. If you go back and look at the literature, you'll see that costs of tests started being a problem in the 1970s and people started focusing on it at that time, and as they focused on it, the cost still went up. I remember sitting in a meeting once, a roadmap activity, where the test group had decided that the future cost of a mainline ATE system for microprocessors was going to be $20 million. And Alex d'Arbeloff [the late co-founder of Teradyne] was there, he was promoting that, I guess. He believed that $20 million was the right number. Well, obviously, it never happened, so we had some impact by trying to reduce the cost of tests. And a lot of things have been done. Anyplace where the incorporation of BIST [built-in self-test] has a cost of silicon area that is less than the cost to test, then it is a candidate to be adopted. And as the number of transistors goes up, the penalty for BIST transistors gets smaller and smaller in most architectures, so that has worked, the incorporation of Built- off Chip Self-Test. There have been a number of projects, and there are some actually still heavily engaged today, where you take the same engine and the same technology, you put it in a chip that's a test chip, mount it so close that its drive and input channels have the same kind of communication characteristics as an on-chip transistor, and so now you don't have to use the energy in that particular device. You can use a more general purpose BIST* engine to drive it, and people have done that successfully -- greater parallelism in test, and we'll come back to that again in a moment. Strip test is one of the mechanisms for greater

parallelism in tests, and it has many things to recommend it, and yet, for some reason, it doesn't own the market. And since there's somebody in the room that knows more about it than I do, I'm not going to say anything else. Increased pin count and probe cards, now, that's a really interesting one and I have something else to say about it later, but clearly people have gone past the lunatic fringe limits of trying to do pin count and probe cards. Multi-side handlers, reduced NRE, and that comes from automatic test program generation, reusable hardware, independent tests, and more. Despite all these things, test costs continue to rise. The rate of rise was arrested, but test costs continue to rise.

So what more can be done? Well, we might reduce burn-in by using acceleration techniques, and that means higher temperatures, shorter times, and identification of faulty die by the analysis of observed parametric changes. Some work has been done on that, the work is very promising, I think eventually we'll see a lot of the burn-in go in this direction because it's an enormous cost savings, and it saves one other thing. Burn-in will weed out weak die and, in the world where transistors wear out, it will wear out the good die. What percentage of the wear-out you use is a different question, but it will definitely wear out the good die. And here is one that maybe most of you haven't thought too much about -- changes in test quality requirements. Anybody heard about that, changes in test quality requirements? Well, I'll show you what we decided after a meeting with a test technical working group of the roadmap this year -- and reduction in test interface cost. So changes in test quality. Well, the test paradigms are changing, and they're changing because we have a billion transistor ICs. We incorporate 3D IC structures and so the ability to prove that we have a known good die gets more difficult, particularly when you have to make a contact that has good power integrity issues to a die that's only 25 microns thick. SOC architectures, SFP

sub-mitigation wafer thinning, smaller contact paths with tighter pitch, and these changes are all driving increases in costs of tests. And you can't get away from them, at least I don't know how to get away from them. If you didn't have to deliver power, then you could just capacity of a couple for logic and memory, and you'd be okay. But you can't capacity of le couple* your power. So the concept of known good die will be displaced by probably good die for very complex systems, and that was a serious result of a lengthy technical discussion between the parties involved, but it is okay. There's another factor just to keep in mind -- there is a cost of shipping a bad die, and there's a cost to test, when they start crossing over, then you start re-thinking your economics and so maybe you do less tests, you take more risk of a good die, and people are doing that already.

Economics will limit penetration of known good die as test access becomes more limited and probing is limited by probe damage. You've got a very thin die, you've got small contact pads, they're very tight pitch, and now you have to ensure power integrity with power and ground connection. We can find a way to get around the quality of the connection for logic and memory, but we can't get around it for power integrity. So probably good die will be enabled by system redundancy. They'll be enabled by a continuous test while running dynamic self repair, graceful degradation, and, you know, that's something maybe you haven't heard too much about in the context of semiconductors, but we're just following the human body model; mostly, we don't go out all at once, they're kind of a graceful degradation. Looking around the room, I can see I'm not the only one that's probably experiencing that.

Now here's a really big one, and you probably know this one, but test interface cost issues are a huge issue, they cost way more than the tester -- by multiple times, more

than the tester. And the interface cost is going to rise rapidly as the bandwidth goes up, and there's an arbitrary selection here, which I actually took from a number in the text of one of the roadmaps of 2 gigabits per second. You can pick whatever arbitrary number you want, it's still going to get more expensive because you have several issues that you're fighting against, one is the drive transistors don't have any power, so they can't drive much signal channel impedance, and the other thing is you've got to have special wiring. Interface cost increases with a number of sites to the point where maybe going to more sites actually increases the cost of tests with more parallelism, rather than decreasing it. High frequency and high parallelism will become impossible for obvious reasons; we can't drive the channel of the tester. And time-to-market issues become more and more a problem with higher and higher parallelism and interface. Interface costs include the cost of interface electronics, the cost of test sockets with spares, the cost of probe cards with spares because, if you don't have the spares and something goes wrong, your entire test cells shut down and your productions shut down. And it also includes the cost of maintenance, so if you have to clean your contacts on a regular basis, then you've got to have spares just to clean the contacts, and sometimes they break -- hopefully, they break in a way that they can be repaired, but as the parallelism gets higher, they get to a point where you really can't repair them. And some of the manufacturers now say, "Well, if you only have four pins broken, we'll repair it, but if you have five, you have to throw it away and buy a new one."

So this is a projection of the ATE interface costs, and as you can see somewhere around here, the projected interface cost goes above the ATE cost. The cost reduction from the trend is a theory -- nobody has done it yet, it hasn't been proven yet, maybe it works, maybe it doesn't. But, for sure, the trend is going to increase with reduced

product lifecycles because you're going to have to do new interfaces as the product lifecycles change. And so we're going to be in a position where the interfaces cost more than the tester, even before we account for the fact the interface has to be changed many times during the life of the tester.

Okay, now, what does all this have to do with test-in-tray? I have to come to that conclusion really quickly because I'm about to use up my time. And I don't want Jim to get upset about it. So what does it have to do with test-in-tray? Well, these are the elements of manufacturing tests, and they're the ones defined by TRSO* -- I just decided to use somebody else's, and I won't go through them all because what I really want to point out is, the potential exists for all of the ones that are in yellow to be improved by the characteristics of tests-in-tray. Now, that potential has no guarantee that any of them will ever be realized, but it is a guarantee that there are some characteristics that will have benefit in each of those if they are not some offsetting disadvantages that get pointed out. So cost reduction occurs through standards in handling die in packages, lower cost for the functions of both the program and the handler, reduced NRE* for new designs, and particularly with regard to sockets and probe cards because we have a standard location where our contacts will be located, even though their pattern will be different, reduced costs for burn-in, increased through-put for indexing in DAT, and reduced cost of the interface, which is already included potentially up here under this one. Finally, reduced time-to market. So, all of these things are factors that test-in-tray may bring to us. And since most of us don't know much about test-in-tray yet, I presume that, from our discussions this afternoon, we can decide if this potential will be met.

The potential benefits of test-in-tray will not be met unless changes are made

in the test equipment, changes are made in the contactors, and changes are made in the other interface components, so that the advantages of test-in-tray are not hidden by the other characteristics of the test cell. Thank you very much. [Applause]