# ARM Holdings plc

*Special Call*

*Friday, December 16, 2016 4:30 PM GMT*

# Call Participants

**EXECUTIVES**

**Jem Davies**

**Philip Sparks**

**ANALYSTS**

**Ambrish Srivastava**
*BMO Capital Markets Equity
Research*

# Presentation

**Philip Sparks**

Thank you, everyone, for joining our call this afternoon. I'm Phil Sparks from ARM's Investor Relations team, and I'm joined here today by Jem Davies, ARM Fellow and VP, Technology; and Ambrish Srivastava, Managing Director and Equity Analyst at BMO Capital Markets.

Today Jem and Ambrish are going to discuss machine learning in ARM powered client devices. And earlier this week, Jem published a 20-minute presentation on this topic on ARM's YouTube channel.

Today's call will discuss that presentation in more detail. But if you haven't seen it yet, don't worry. You can find the presentation by typing machine learning on ARM powered client devices into YouTube's search bar. This will be a listen-only call. So if you have any specific questions, you would like answered during the call, please submit them using the questions box on the webinar control panel. If we don't get a chance to answer your questions on today's call, we will follow up with you afterwards.

So without further ado, I'll now hand over to Ambrish.

**Ambrish Srivastava**
*BMO Capital Markets Equity Research*
Thank you. Thank you, Phil. This is Ambrish with BMO. Jem, thank you for making the time and pleasure to talk to you again. This is a pretty timely piece you put up on the ARM channel. As we think about artificial intelligence, AI, machine learning, deep learning, a lot of terms that are now topics du jour. Your experience over the years is great to help us gain some insights and perspectives.

# Question and Answer

**Ambrish Srivastava**
*BMO Capital Markets Equity Research*

So first of all, let me start out by a question on what is ARM seeing in the marketplace in terms of a -- whether it's a greater adoption of AI, and machine learning. So if you could please share with us a perspective on over the last year or so, what is ARM seeing in terms of the inbound requests as well as working with you partners? Whether there is a greater adoption, greater engagement around this topic?

**Jem Davies**

Yes, thank you very much. So it's perhaps helpful if I just start off by defining a few terms, make sure we're all on the same page. So artificial intelligence is sort of an all-encompassing terms. It's -- trying to get computers to apparently be smart or to think. Of course, it's interesting because [ph] once it actually starts working, people will stop calling it AI, they would just call it computing. Machine learning is a similarly large field and a very old one. There's been lots of work being done in machine learning over the years. What is perhaps most interesting over the last few years is the application of so-called deep learning and neural networks. So we've seen a couple of very high profile things like the computer vision, teaching computers to recognize images where the application of deep learning to that has got computers better at recognizing objects in pictures than people now. Better correct detection rates, lower false positive rates, and a couple of other very-high profile public things, like the application of deep learning to voice controlled services on handsets. And even the successful beating of chess grandmasters and Go masters by computers. So this has caused everybody to get a lot of attention in this area. But of course, like all sudden overnight successes, there's usually been the 25 years of hard sweat and labor before that comes up to that point.

So in terms of ARM's marketplace, it's really all about our partner saying very much, "Hey, about this machine learning thing, what does this mean for us?" That they're very keen to be educated, very keen to be led by ARM. And they want to know what this is going to mean for ARM's R&D efforts. What products we're going to be producing which will be affected by the trends that we're seeing for deep learning.

**Ambrish Srivastava**
*BMO Capital Markets Equity Research*

Okay. So kind of related to that, what is ARM doing different from a development perspective to address these new opportunities, these changes in a qualitative way or any metric you could provide? Where is the shift, if you will, on the R&D dollars to address, what seems to be a pretty large opportunity over the next few years?

**Jem Davies**

Yes, we agree. We do think this is going to be a very large opportunity. I do think that machine learning altogether is probably going to be one of the biggest shifts in computing that we'll see in quite a few years. I'm reluctant to put a number on it like -- the biggest thing in 25 years or whatever. But this is going to be big. It is going to affect all of us. It affects quite a lot of ARM, in fact. The CPU group will be looking at the workloads, the neural network compute frameworks. And using that as examples to our performance analysis and benchmarking efforts to say well, what could we do to our CPUs to make them better at performing on those workloads. Equally, of course, once you step out of the CPU domain into something like the GPU area, then for us it's all about non-generic workloads, so GPUs, graphic processing units. And of course, there's a clue in the name there, are specifically, chosen to be very good at graphics. And it turns out the way that graphics is done efficiently is we have very parallel [ph] based workloads. We're able to do lots of things at once. And graphics is, in fact, very computationally intensive. There's lots and lots of arithmetic being performed in graphics. So it turns out that GPUs are actually very efficient at performing these sorts of workloads.

Now for years, we've had very highly developed benchmarking content analysis, performance analysis and virtuous feedback loops from that into GPU design, because we are trying to produce a processor here that is specifically tailored to a type of computation. Graphics, it is a digital world. The pictures on the screens get generated from GPUs. It's all 1s and 0s. So there's lots of computation in graphics. And what we're trying to do is to produce GPUs that are very good at executing, and by good, of course, I mean, efficient, high-performance, low-power. And it's an extension of our work in that area, which is getting us to look at the neural network and machine learning workloads to analyze those in our GPUs, and to inform our future road map. What should we be doing to the design of our GPUs to make them even better at running machine learning workloads. And indeed, is there a place in our roadmap for future products, which are more specifically tailored to this.

## Ambrish Srivastava
*BMO Capital Markets Equity Research*

Okay. In your presentation that you provided us with. You went through and you talked about certain use cases where a device might be the right place to use machine learning than on the server side. And most of the discussion on machine learning has really centered around the server side. So if you could please help us understand under what cases is it more useful to deploy or use machine learning on the device side as opposed to on the server side?

## Jem Davies

So again, with your permission, what I'd like to do is just sort of step back from the question lightly -- slightly and set some background. So machine learning has been very successful on the server side, initially because the server side tends to be where the data is. So if you looked into the presentation I talked about, the difference between training and inference. And the process of training neural networks, in particular, is best done where people have vast quantities of good quality and annotated data. So if you are one of the big service providers, who has access to vast amounts of our data, then you are in a good position to do this training of those networks to teach the networks to perform the task that you're trying to do.

Now of course, having trained a network to do a thing, it's actually quite a separate process to do the inference. And you say, well, okay, I've now trained this thing. Now I want to run this thing and have it identify pictures of cats or perhaps something more useful. But the place that thing might be done, might well be more convenient to users, if it was on devices that they have in their hands. And more than convenient, there are some very fundamental technical reasons why some of the things that you would want to do would be best done locally on device without reference to servers. If we look at automotive, for example, if you look at ADAS, the safety critical systems, we can't have those being reliant upon a connection over the Internet to a server. Your mobile phone signal might drop, WiFi might drop, whatever it is, whatever your connection mechanism is. We know there are cases when that connectivity may drop. And we can't have a car's ability to recognize another car in front of it being dependent upon that connection. Equally, even if the connection is there, there are latency issues. So if I were, for example, to collect all my data, send it all the way up a server somewhere in another continent, have it processed there and then come back, that is all going to take a finite length of time. Even with radio waves traveling at speed of light, the distances involved are not your friend here. We need to respond to certain things, particularly safety critical things, but also even for user experience criteria. We need to respond to them pretty quickly. So latency is another issue.

Then when you talk about a future in which such capabilities have been scaled out to billions, literally billions of devices, not just mobile phones, but also devices in the Internet of Things. Then the bandwidth available for these communications come into play here. That bandwidth may not be available, simply because there's so much device data being thrown around the place that we've broken the Internet. Or it may manifest itself as power because the power it takes to transmit data is often greater than the power it would take to process it locally. So again, there would be strong reasons why you would want to process it locally on device rather than send it up to a cloud server.

And then finally, there's security. As machine learning becomes more pervasive with -- in devices, we will find it being involved in things that we care about. It may be related to our personal data, perhaps related

to our health data. If you had a server that is trying to detect whether you are in fact, having either a panic attack or a heart attack, you probably don't want your personal health data being transmitted around the world. Or if you did, you'd be very concerned about the security associated with that. And there are multitude of examples there where security -- you would feel an awful lot better about if it stayed locally on device and weren't transmitted around the world up to a server.

So what we find particularly, in summary then is to divide the problem up into machine learning training and machine learning inference is useful. It helps explain the problems faced [ph]. And then looking at the use cases where you might really want that inference being done on device side, not on server side. It's not to say it's a battle. There are clearly use cases for both. But we think that there will be a predominance of use cases for inference where you will want that done on the client side locally on device.

**Ambrish Srivastava**
*BMO Capital Markets Equity Research*

Good point. And I think to your point on the ADAS opportunity, it's hard to imagine that -- oh, wait a minute, we don't have connectivity, and we need to recognize what's right in front of us. Let me just bring it back to ARM. So when you think about machine learning, and the billions of devices that are out there [indiscernible] that's out there with ARM technology. How does ML change the opportunity for ARM? Taking it beyond handsets. We'll start with handsets and then how do you see it expanding beyond that?

**Jem Davies**

Well, we see, of course, handsets is a huge area because everybody's got one. And every year they get more capable, and it's often the place where new capabilities are first shown. Because it's actually relatively easier to put a new capability into an already very complex and capable device. You can sell it as next year's model. And everybody's going to start carrying it around in their pocket, as opposed to introducing a new form factor device. And they say, "Oh, well, I've got to carry 2 things now." Or maybe I'll think about that before, I'm really convinced I need it. So handsets is the most amazing market for introducing new technologies. And we see that as -- we've seen mobile being the first area in which so many things are showcased. Machine learning changes the opportunity for ARM, I think, in several ways. The capability of machine learning, even inference-only on device is going to require more compute capability. It's going to require compute capability of a particular type to be done in a power efficient fashion. That's going to require us to produce more capable processors. So we have an opportunity there to create new products and more capable products in that space. And then as we see that fan out, as I say, we see a start of these capabilities in handsets. But that's going to fan out because we've seen what becomes first groundbreaking in mobile phone handsets. We see progressively spread out into the world of other consumer electronics devices. And indeed, then the creation of new factors [ph] of consumer electronics devices. The so-called smart connected devices that you're seeing in your home.

And so what does this mean for us in market terms? Of course, it means we have an opportunity to sell into more devices. We have an opportunity to sell more capable products into those more devices. And we've an opportunity to add value to the products that our partners create. And so we look to increasing royalties there. If you look at the end markets, it's not just going to be mobile. Of course, there is no market in the world like mobile as we've already talked about. But self-driving cars have had a lot of press, autonomous self-flying drones is an area that is almost entirely powered by ARM at the moment. Security cameras is becoming an area that is basically -- all of the security cameras in the world are basically now being replaced by smart cameras, by cameras that are already doing analysis of the images that they collect. And looking at being more smart with what they transmit, what they record. And in safety terms, picking out features that are relevant to public safety or whatever. One of the problems with cameras -- well, it depends on -- if you're a disk drive manufacturer, of course, it's great. A high def camera will transmit enough data to fill up a modern standard 1-terabyte hard disk drive in about 24 hours. But of course, that's not really sustainable. So what we need to do is to replace all of those surveillance cameras in the world with smart cameras, that only record that which is important. In a very similar way to the way that human beings do. If you say to me what changed in this office in the last half hour, I would say well, not very much. Oh, yes -- Ian got up and walked out and came back in again. That's a very, very small amount of data compared to recording a half hour of 4k video.

So we think security cameras is going to be a huge market for the addition of computerization and [ph] machine learning capabilities. And what we're seeing with new use cases for new format devices, such as Amazon's Echo. But also, of course, there are lots of other devices like that. Is that we're seeing the digital assistant taking a new format, a new user interface, a new method of interacting with people that is much more natural to them. Kids, for example, kids love Amazon Echo with the Alexa personal assistant. It becomes -- I've literally heard people say, "Oh, yes, Alexa is a member of our family." They talk about it as one of them. Kids just align with it very, very quickly. They have different set of expectations to us. So I think you'll see a huge change with machine learning capabilities being added to devices to kind of break down that glass barrier between ourselves in the real world and the digital world.

**Ambrish Srivastava**
*BMO Capital Markets Equity Research*

Okay. And I know Alexa is smart and kids are smarter because now I see in my in-box orders that were placed by Alexa, and it's my kids placing those orders.

**Jem Davies**

I believe [ph] it will have to be controlled.

**Ambrish Srivastava**
*BMO Capital Markets Equity Research*

Maybe just stay with smartphones, and then we can go a little bit more technical that I wanted go into. But ARM has made some acquisitions and many and some of the -- and I want to -- don't want you to name any customers, but some of your ARM's partners have tremendous capability themselves. So is the model we should be thinking about is smartphone makers are assuming 100% of them are using ARM CPUs and graphics close to 80%, 90%? Then some smartphone makers are going to be differentiating via their own machine learning capabilities, and then for those that don't have that, what is ARM bringing to the table on the software side?

**Jem Davies**

Yes, sure. So I mean, as you will understand ARM's traditional business model is that of working to supply the partnership with the things that the partnership requires in order to make great compelling devices that they then sell and make money out of it. That traditional marketing economics doesn't change. Machine learning doesn't break any of that. It provides something of an inflection, it means that we all have to be providing devices with these new capabilities. And obviously, ARM is spending a lot of its R&D [indiscernible] on the work that I talked about earlier, adding these capabilities to our CPUs, our GPUs and other devices like computers, special purpose computer vision processors, and others yet to be produced. And the traditional [indiscernible] by economics apply just as much for this as it ever did. And what we find is that this is actually a significant piece of engineering effort and research that we're doing that costs a lot of money. We have to employ a lot of smart peoples to do it. And so if you're not shipping a lot of devices, then it's not going to make sense to do your own. But even if you are shipping a small number and buying from us, you're going to be pushing us, and you're going to be saying, "Well, look I need these capabilities and I can buy this from you or I can buy this from anybody else." It is a competitive marketplace, it always has been. We are continually pressured extremely hard by our partners to provide more capabilities and more efficiency and to provide things more cheaply. None of that's going to change, I'm confident. It's always going to be a tough marketplace for us to have economic pressures on us to which we respond and produce great products. If we don't do that, we'll fail. But obviously, I'm confident that we are doing that right. And we're supplying our partners with what they need. Some of them clearly will have specific requirements that they think that they can better provide by their own internal engineering efforts. Of course, that is the case today, and that will continue in the future.

**Ambrish Srivastava**
*BMO Capital Markets Equity Research*

Okay. Let me turn a little bit into the technical side of things starting with and I say a relatively straightforward one, precision. Help us understand the difference between floating-point 32

[indiscernible]. And then in your presentation you have talked about int8. And specifically, you mentioned that going from 32 to 8 bit integer code there is a power consumption saving that you entail, which then helps on the client side. So part a, a quick primer on precision. Part b, what kind of power savings do you get to support the claim that from 32 to 8 there is a big power saving. And then hence, easier to use on the client side as opposed to on the server side?

**Jem Davies**

So integers are whole numbers. So 1, 2, 3, 4, FP, floating-point, 3.4, 5.6, et cetera. The precision, also called the width of the numbers illustrates the scale of those numbers. So an 8-bit integer can code the numbers from 1 to 256. A 16-bit integer will go from 1 to 65,000. Floating-point numbers have a huge range of numbers that they can represent. But the width of that floating-point number usually affects most directly the precision with which it can -- sorry, the accuracy with which it can express that number. So if you think the number of places after the decimal point is what comes up in the difference between the lengths of the floating-point numbers. Standard precision in floating-point is generally considered to be 32-bit with FP 16, 16-bit floating-point numbers generally referred to as half precision. And why do we care so much about this? Well, first is storage. Obviously, you can store twice as many 16-bit numbers as you can 32-bit numbers. And memory is still an issue. It's less of an issue than it used to be, but it's still relevant. It does very definitely affect bandwidth, bandwidth equals power, as we've talked about earlier. So if you're transmitting or storing your numbers as smaller numbers, then you're actually using less power to store those numbers. And then internally, in the very insides of our processors, the ALU, the arithmetic logic unit, it is the square of the width that affects the amount of power. So for me to do a 16-bit floating-point calculation is probably 1/4 of the power than a 32-bit floating-point number. And similarly, down with integers as well.

So why is this all relevant? What tended to happen with the initial development of these algorithms is as I said they were developed by the people who had access to vast quantities of data. And of course, the people who have vast quantities of data tend to have vast quantities of hyperscale servers. And they all have double precision, single precision floating-point as standard. And nobody thinks too much about it. When you come to transfer those trained neural networks down on to devices, several things come into play: first, the size of the code and the size of the storage of those neural networks. And then the actual efficiency with which they're executed, and the power they consume whilst being executed. So at this point, when the device guys get into play, they say, well, hang on, I care a lot about this. Do I really need full 32-bit floating-point precision? Can I recast this algorithm to run in a smaller width of data type? And the answer to that is yes, with a surprisingly small reduction in the accuracy of the algorithms. So for example, nothing is ever perfect. And an image recognition algorithm might only be 98% accurate at recognizing polar bears. And actually if you reduce the width of everything and compress the neural networks, you might end up with something that's 97.5% accurate or something close to that. And a lot of the work that's gone on in the last year, including most publicly from Google with their TensorFlow framework is production of a whole set of tools, which take a neural network, which has been trained on a server. And then compressing the network itself and reducing the precision of the data types used in that network to make it more meaningful to running on devices where memory is short, bandwidth is power and power equals battery life.

**Ambrish Srivastava**
*BMO Capital Markets Equity Research*

Okay. Turning over to compute power. What is the right way to think about compute power, either in terms of gflops, int8 or otherwise in training. And more importantly, in inference where speed and efficiency are more important. So for example, what I'm trying to get to is, how does an ARM CPU and a Mali combo compare with x86 CPU and a GPU combo? Are there any benchmarks you can provide us, which would help us understand where a ARM combination, for the lack of a better word, would stack up against a competitive instruction set in a GPU [ph]?

**Jem Davies**

Sure. So gigaflops is 1,000 million floating-point operations a second, which is a number that often gets bandied around when trying to measure the capability for computation. As I just talked about,

actually an awful lot of those operations on inference on device side are now being moved from 32-bit floating-point operations to smaller width floating-point operations and increasingly to integer operations. So increasingly, the gflops metric is getting a little bit confusing. And what we now need to do is with knowledge of the algorithm you're trying to run, and the type of data that you're trying to run it on, how many operations per second can your device perform? So we now see GOPS, giga operations per second or even tera operations per second. And the operation in that number, of course, isn't immediately defined. You have to know that your tera ops per second are the same tera ops per second as you are comparing to your competitors' tera ops per second. And so actually here the comparison unfortunately, does get a little more involved. You have to know what it is you're comparing like with like.

Now in terms of benchmarks, there are some fairly commonly used ones on the image processing side. So one very commonly used example of neural networks and machine learning is to do scene recognition to recognize the objects in a scene. I showed a demo of an application doing that in the slide deck. There are several academic data sets, and therefore, benchmarks in that. And the tera ops per second doing that or image recognitions per second doing that have become something of a common benchmark to do that. And of course, once you have something like operations per second or recognitions per second, the next thing people care about is operations per second per watt or recognitions per second per watt. Because in nearly all client devices, the capabilities of that device will be thermally limited. With the biggest battery in the world, it's still only going to be able to consume a certain number of watts. And if you are moving towards always-on devices, which are waiting to be woken up by you saying hey, Google or Alexa or whatever, then that always-on capability is going to be one where power consumption is absolutely the defining critical factor.

**Ambrish Srivastava**
*BMO Capital Markets Equity Research*

So which was my -- you answered my related question that I was going to ask is, where does the power come into play?

**Jem Davies**

Yes, as I said, your high-end handsets are universally thermally limited. They can't get used more than a certain power. And even in lower-spec devices and the IoT devices, that we're talking about battery life, of course, would be critical.

**Ambrish Srivastava**
*BMO Capital Markets Equity Research*

And then turning to another one and we've talked about it already a few times on the server side, there is a lot of debate, and we tried to address that in a piece that we have written a few weeks ago, about inferencing versus training. What's the right way to think about the ratio of compute power that is required in the cloud? And then how do you translate that to the mobility side? And what I'm referring to is the ratio of compute power between inference and training.

**Jem Davies**

Well, I'm going to answer your question slightly strangely. The compute power on the device is the maximum that you can provide in that to what budget. And so that is the defining factor. The definition is [indiscernible] within that power budget. On server, of course, it's different. And really there it's as much as you can afford in terms of money. So you will throw racks and racks and racks of server CPUs at the problem until you've spent all your money. So the 2 are not directly comparable in that sense. Certainly, we see training taking a huge amount of time. You might be training a neural network for days and weeks whereas, of course, typically an inference on a handheld device, ideally you'd the answer back in 20 milliseconds or something, within human reaction time. So the total amount of power consumed, compute power consumed is in those cases quite radically different.

**Ambrish Srivastava**
*BMO Capital Markets Equity Research*

Okay. So, so far we've touched on -- not touched, we've talked a great deal about servers, and we've talked about mobility. Machine learning in industrial market. Is that -- does that open up more opportunity for ARM as a part of an acceleration core processor working either independently or more importantly, with an FPGA supplier such as Xilinx, which has been very public about using ARM since their Zynq introduction a few years ago. What's the right way to think about ARM machine learning in industrial?

**Jem Davies**

So machine learning has been applied in industry actually for many, many years, in a wider sense in machine learning. We're now seeing neural networks being applied in industry. There is a couple use cases that are driving that, one of which is intelligent monitoring and intelligent maintenance. So in the old days, you had a man in a brown coat who knew that, that generator, that 2-gigawatt generator should sound like that. And when it sounds like this, when it sort of starts giving that slight screech he knew there was something wrong with it. And what we're finding is that actually you can train a machine to do that. So intelligent monitoring, both acoustic and other forms of vibration is proving to be extremely important. Because the last thing you want to do is take one of these big machines down for maintenance when you don't need to. The last thing you want to do is have it go catastrophically wrong because you failed to maintain it. So we find industrial to be a very interesting application space, particularly, the proliferation of connected devices, relatively cheap devices that are doing the sort of monitoring of machine health that is now becoming very easily available to big plant [ph]. And I was at a conference the other day where the -- one of the panel speakers, I was on a panel, one of the speakers was talking about precisely this application where they were monitoring the power distribution network of the entire United States. And they were feeding this into machine learning systems. And extracting all sorts of data from it that they previously didn't know was there. And that data is becoming to be an asset. It is being monetized by them in various ways.

Not just heavy machinery, but also robots, of course, being extensively used in industry these days. Most of them still bolted to the floor, but increasingly now more autonomous. You will have seen pictures of robots maneuvering around Amazon distribution centers and things like that. But of course, [indiscernible] example. There's a lot of machine learning going into those devices. They need to be capable, they need to react to the things not being properly straight on the shelves. They need to react to people walking in front of them. All of the sort of things you can easily imagine. They need to be trained to do that. And as we are present in so many of the microcontrollers and the processing devices contained within these sort of new devices, that's becoming a big area for us.

**Ambrish Srivastava**
*BMO Capital Markets Equity Research*

Okay. In the interest of time, let me -- I have 2 more questions. One referring to an example you have, keyword spotting on phones on Slide 17 of your presentation. A, you mentioned looking at use case for future IP products. So how is keyword spotting done today on flash of phones [ph] with an always-on capability? And then if this were to be implemented using ARM IP, would that require a separate hardware for that always on? Or what would the implementation look like?

**Jem Davies**

Usually, the always-on electronics is kept in a separate power domain. So when you're trying to control power and energy consumption, you always want to turn as much off as you can possibly get away with. So regardless of the processor that it's actually running on, where such as a Cortex-M3 ARM microcontroller, you would have that running in this separate power domain. What you might have is in something that's analogous to the big little CPU thing is you might have a little microcontroller saying, "Well, I think I heard some speech that might have had a G [ph] in it. Tell you what, I'll record it. I'll wake up the bigger one and I'll pass it on to the bigger CPU for him to determine whether somebody said hey, Google or not. And so regardless of whether that's using ARM hardware IP or not, I would say you're always going to have them as somewhat functionally separate even if it's actually in the same chip. You're probably going to want to a separate processing unit doing that work because you're going to want it always powered on when everything else is powered off.

**Ambrish Srivastava**
*BMO Capital Markets Equity Research*

Okay. So putting it all together, and coming back to the title of this presentation that we had, which was when does my smartphone become smarter than I? So when do we see all this come into fruition, whether it's a smartphone smarter than I. Or my car that I don't have to -- I can be lazy and not touch anything and have a fully automated level 5 ADAS system powered by machine learning?

**Jem Davies**

Well, of course, today some things are already smarter than you. Sorry, smarter than me.

**Ambrish Srivastava**
*BMO Capital Markets Equity Research*

Smarter than maybe a lot of people.

**Jem Davies**

So for example, you would never these days have your eyes operated on by a human being wielding a scalpel or mostly wouldn't, except in very rare cases. You would actually have a machine doing that, which is obviously incredibly more precise. It doesn't have bad days and its hands never shake, which has been taught by the experience of lots and lots of humans. Even if you don't have a completely autonomous car, nobody would think about having a car these days that didn't have an airbag that exploded when it detected a force of more than whatever it is g. And applied the antilock braking system when it sensed that the wheels had locked up. So I think what we'll see is the introduction of increasing levels of smartness in increasing areas of capability. My smartphone today, pretty smart. I can use it to connect to services like Amazon's Alexa or Google or Apple's. And I can ask it questions that, there's no way I'd know the answer to. So I ready have access to considerable levels of smartness more than I have. I what I think you're going to find over the years. And I'm not going to make a prediction of the absolute sort of date of the singularity when my smartphone is smarter than me. But I think you will see huge amounts of capabilities being added into the devices around us. And I think there is no limit to what these things can do for us to make our lives better and easier and safer and more enjoyable. Machine learning has the capacity to affect pretty much every area of computing around us. I think it's an incredibly exciting time to be involved.

**Ambrish Srivastava**
*BMO Capital Markets Equity Research*

Okay. Great, Jem. Thank you, thank you very much. This was very helpful and I'm sure listeners who have dialed -- who have logged in would find it very useful as well. Phil?

**Philip Sparks**

Thank you, Ambrish, for leading the call. And...

**Jem Davies**

Thank you very much.

**Philip Sparks**
And thanks, everyone, for listening. And as a reminder, if you'd like to learn more about the topics discussed today, you can find Jem's online presentation by typing machine learning on ARM powered client devices into the search bar on YouTube. And if you have any follow-up questions for Jem, you can e-mail us at investor.relations@arm.com. And we look forward to seeing you in 2017. Thanks, again, and good-bye.