

## Windows on ARM

In 2010 Microsoft got interested in ARM, because of low-power solutions for server-parks. ARM tried to lobby for years to convince Microsoft to port Windows to their architecture and now the result is there. Let's not look to the past, why they did not do it earlier and depended completely on Intel, AMD/ATI and NVIDIA. NB: This article is a personal opinion, to open up the conversation about Windows plus OpenCL.



While Google and Apple have taken their share on the ARM-OS market, Microsoft wants some too. A wise choice, but again late. We've seen how the Windows-PC market was targeted first from the cloud (run services in the browser on any platform) and Apple's user-friendly eye-candy (A personal computer had to be distinguished from a dull working-machine), then from the smartphones and tablets (many users want e-mail and a browser, not sit behind their desk). MS's responses were Azure (Cloud, Q1 2010), Windows 7 (OS with slick user-interface, Q3 2009), Windows Phone 7 (Smartphones, Q4 2010) and now Windows 8 (OS for X86 PCs and ARM tablets, 2012 or later).

Windows 8 for ARM will be made with assistance from NVIDIA, Qualcomm and Texas Instruments, according to their press-release [1]. They even demonstrated a beta of Windows 8 running Microsoft Office on ARM-hardware, so it is not just a promise.

How can Microsoft support this new platform and (for StreamHPC more important) what will the consequences be for OpenCL, CUDA and DirectCompute.

The APIs

There is an image of Microsoft Azure, since that comes most close to the modular OS that Windows 8 will be. Moreover the image explains well what Windows-on-ARM will be: replace the word "Windows Azure" by "Windows 8 on ARM" and you see that all the software on top still runs - according to the picture. We speak in terms of functionality, not in byte-code. It should not matter if you say something in French, Spanish or English, as long if the same vocabulary is used. C-code can be compiled to ARM since a long time in Linux (Microsoft, say "Thank you, Linux"), so it is only a matter of using the same APIs for the rest.

## Recompiling and multi-platform

Wine is an example of how a piece of software can run on another platform, as long as the APIs are known [2]. If Microsoft succeeds to have the same Windows-library-interfaces on their new Windows8, then all software can easily be ported.

Apple has shown Microsoft how to change the processor in their machine, when switching from IBM POWER to X86. Software-packages contained both a POWER-binary as a X86-binary, support for the old system was slowly dropped. We could expect something similar here: dotNET would be the defacto-standard for Windows8-software, while VisualStudio could have easy packaging-options for multi-platform msi-files. It completely depends on Microsoft's decision for support old software; after a big clean-up with the introduction of Windows Vista, chances are high there will not be another clean-up to keep business-partners

happy.  
.NET

Microsoft has actually prepared for more flexibility for a long time by introducing psydo-compiler and Java-alternative dotNET, which could replace the Win32-API, which has been built up its history since the 80's. All software that is built with dotNET can run on ARM; the theoretical proof is in the Mono-project (dotNET for Linux) [3], which runs (simple) dotNET apps on Nokia phones. This could mean a fast acceptance of Windows on ARM. Same is for DirectX, which is already integrated in dotNET.

Windows 8 being modular, we can expect support for Java, OpenGL, Flash and other alien technologies be an installable extension. The default being Microsoft's APIs, but easy to install third-party alternatives to avoid anti-trust cases. And what's wrong with that? Google used this most-users-stick-to-the-defaults-trick with their services in Android, without somebody sending an army of lawyers.

## Innovation and Intellectual Property

ARM is important for the IT-industry, since in the ARM-world the technology (Intellectual Property) and the manufacturers are mostly split, there is room for much more innovation. For example, if on X86 the IP was maintained by IBM, then for instance NVIDIA could have bought a license and start competing with Intel and AMD. As we have seen with Intel's Atoms, a company is on its best when the competition is innovative too. But what the entrance of these big companies into ARM actually does concerning Intellectual Property to gain or protect their market, we just have to see.

The ARM-world could be drawn out of balance now with X86 in their neck, but Microsoft could even help protect the ecosphere. I cannot say anything about it, but just want you to know it is different from the X86-world and there could be some sort of a clash of cultures. Please comment, if you have insights.

## DirectCompute, CUDA and OpenCL

Since I expect Windows 8 to be strongly dotNET-based, I think OpenCL and CUDA will have a real confrontation with DirectCompute not earlier 2012/2013. GPU-vendors to support their technique when they can sell more hardware, but it will take a while until GPGPU is adopted on Windows. Another reason for DirectCompute-adoption would be the Xbox 720, since the 360 does not support GPGPU. But that gaming-machine will probably be released in 2012, just like Windows 8. All Microsoft can do is doing their preparations well for that time.

So CUDA and OpenCL have almost 2 years to get their part of marketshare. That's a very long time in IT and is of strong influence on this forthcoming battle for market-share.

Microsoft self and all its ARM-partners NVIDIA, Qualcomm and Texas Instruments are members of the Khronos OpenCL-group, so repeat my statement: Microsoft should add support for OpenCL-kernels in DirectCompute, before ending up with a very expensive product without a big market-share.

### References

- [1] [Microsoft press release about Windows on ARM](#)
- [2] [Wine on ARM](#)
- [3] [Mono on ARM](#)