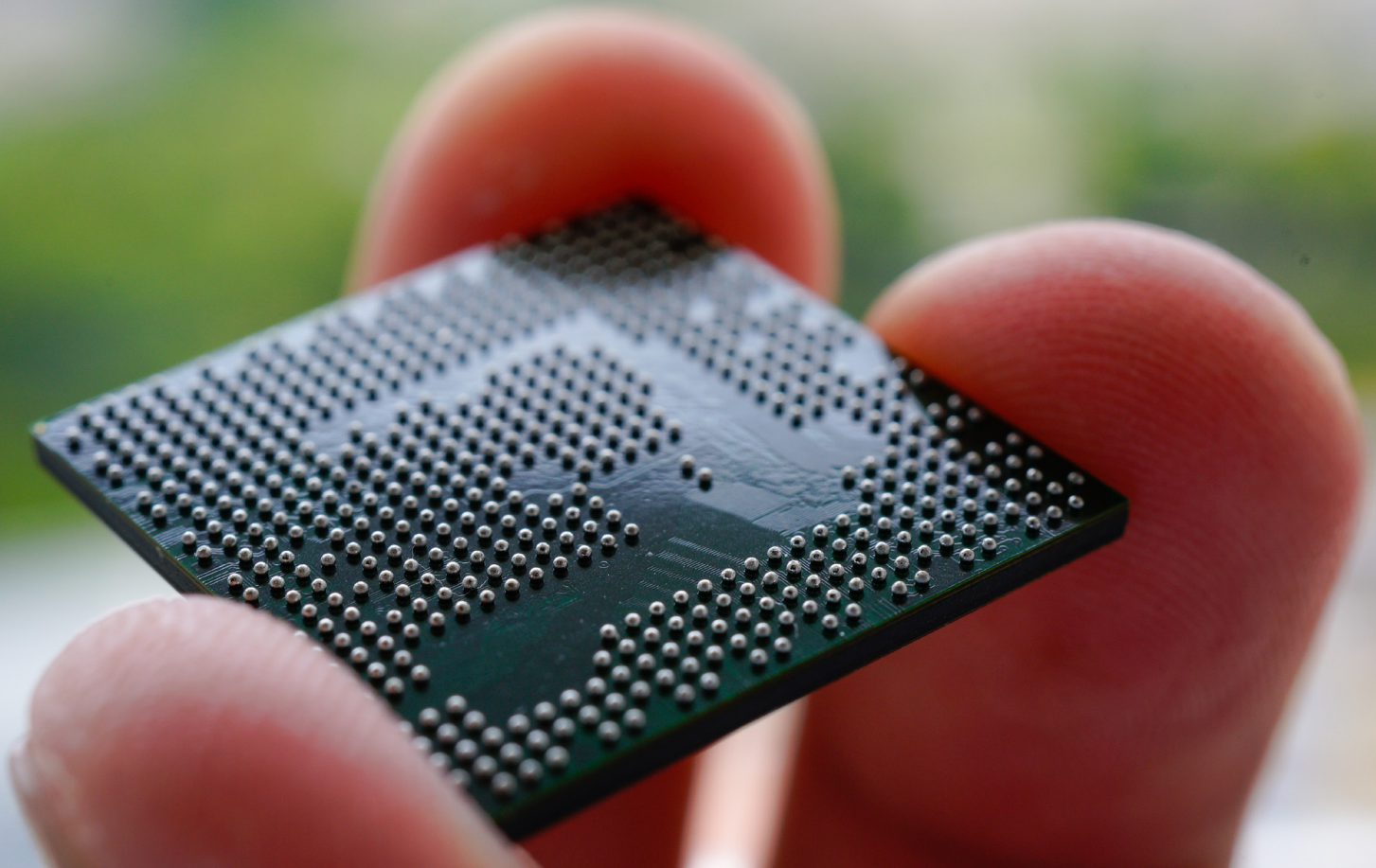


# Choosing a CPU

Is the i.MX 8 right for your project?



Till Adam | Chief Commercial Officer



## Selecting your hardware is among the earliest decisions you need to make when building an embedded system.



Make sure you consider your current project as well as your company's future roadmap when selecting hardware.

### Choosing a CPU

When building an embedded systems product, among your earliest decisions is the choice of hardware. More specifically, on what CPU should you base your design? Today's system-on-chip processors have a huge array of resources to contribute to your solution: multiple cores and on-board DSPs, graphics engines and display controllers, peripheral support and connectivity interfaces, and more. Because a new hardware platform entails a costly investment in hardware engineering, electrical design, and software development – as well as being the basis for future product spin-offs – it makes sense to consider your hardware selection wisely.

If you're like many of our customers, you may be wondering if an i.MX 8 chip could form the core CPU of your product family since NXP has at long last brought the power of 64-bit processing to embedded with the i.MX 8 series. With an i.MX 8, your design would have enough horsepower to last for many product generations. Yet the i.MX 6x family, a predecessor of the i.MX 8, has been a venerable and trusted CPU for many projects over many years. With so much of your company's future riding on your hardware selection, deciding between the two is not easy to do.

We've learned some hard-won lessons in using i.MX 8 silicon to bring customer projects to life and have helped customers determine what's the most appropriate CPU for their current product and future roadmap. We'll share with you some of our CPU choice considerations, helping eliminate some of the unknowns and hopefully clearing away some misconceptions in the process.

### 1. Processing power

Clearly one of the biggest updates between the i.MX 6 and i.MX 8 families is in processing power. The i.MX 6 family uses a 32-bit architecture with a clock-speed between 528 MHz (for the ULL) and 1.2 GHz (for the i.MX 6Dual/Quad). Although there are dedicated dual and quad core versions, most i.Mx 6 chips have a single core.

## The entire i.MX 8 family has a 64-bit architecture, running between 1.2 – 1.6 GHz with up to four Cortex-A cores.

**The main question is: do you really need all that extra processing power?**

In contrast, every member of the i.MX 8 family has a 64-bit architecture running between 1.2 GHz and 1.6 GHz with up to four Cortex-A cores. In addition, all of the chips have an extra Cortex-M core, several models have an extra DSP; the i.MX 8QuadMax has two additional Cortex-A72 cores. Clearly there is a lot more oomph in the i.MX 8 product line!

The main question is whether you need that extra processing power. If you've got an existing product, look at your CPU idle time with your operating system's performance measuring tools. Chances are good that you're not anywhere near maxing out the existing CPU under most normal situations. Throwing extra cores at a problem only works if software tasks are well-divided in processing load and independent execution. If all of your tasks on one core are waiting for another core's operations to complete, you may need some software redesign more than extra CPU horsepower.

If you don't have an existing product for comparison, it's much harder to tell what your anticipated CPU needs will be. Cases where the i.MX 8 power may be warranted are when consolidating multiple CPUs, running hypervisors for multiple virtualized operating systems, or in high-bandwidth or big-data applications like computer vision or machine learning. For most normal "run-of-the-mill" embedded systems, something in the i.MX 6 family will probably suffice.

### 2. Graphical needs

Few embedded products require 4K resolution – but for those that do, this is a clear-cut reason to use an i.MX 8 since the i.MX 6 doesn't have 4K support. However, 4K has a lot of pixels to manage! Our testing of the performance on the i.MX 8's GPU shows that frame rates for most 3D scenes fall significantly short of what's needed for smooth animation. If your product only requires 2D windows or statically rendered 3D screens this may not be an issue, however really testing with your desired output is the only way to know for sure. The higher the resolution and the more objects needed to compose your screen, the more challenges you'll have in achieving decent frame

## Don't expect every possible corner case to be working until customers use a CPU in unanticipated configurations.



Plan to build in time for troubleshooting with new CPUs to debug and develop workarounds if necessary.

rates, regardless of chip selection. You may need to – as we've done for our clients – perform a lot of optimization to your 3D models, shaders, and rendering pipeline to accommodate higher resolution displays.

Another potential reason to select an i.MX 8 is if you're decoding or encoding video. There are some video codecs that are only supported on the newer hardware: for example, VP8 or VP9 decode. (Best to check the data sheets for a full run-down of supported standards.)

### 3. Product maturity

With today's ultra-complex hardware designs, you can't expect any manufacturer to have anticipated every possible corner case until customers push the product to its limits and use it in unanticipated configurations. You should expect some revisions before everything is completely stable when you're developing with CPUs at the cutting edge.

Building in extra time to understand and accommodate new hardware is a good practice that's independent of any assessment of the i.MX 8 itself. No matter how dependable a company's track record in delivering well-tested hardware, more errata sheets are issued at the beginning of a product's lifecycle than at the end. Of course, the impact of hardware quirks or driver bugs in your project may be nothing, and that's more likely if your design sticks to commonly used features. If you find a problem – you'll need to analyze and understand it, and then hopefully install new drivers, get firmware updates, and deploy hardware or software workarounds.

Given the relatively recent release of the i.MX 8 silicon, it's prudent to consider building in some time for troubleshooting. With development cycles on the long side (two or more years to completion), there's enough time to smoke out as-yet-undiscovered errata that could affect your project. Developing at a rapid six-month pace doesn't give a lot of additional time to debug and develop workarounds if any hardware problems are uncovered.

**Excess heat is only part of the problem with a revved-up chip; it also consumes more electricity.**

**Unless your product design requires a CPU that can fry eggs, expect to employ thermal regulation.**

#### 4. Batteries and heat

Faster clock speeds moving much more data also means your CPU will be generating a lot of heat. Unless your product design requires a CPU that can fry eggs, you should expect to employ thermal regulation. The i.MX 8 runs hot when you run it flat out. A passive heat dissipation design with fins and copper heat pipes will still need software assistance. This is achieved by powering down unused parts of the chip, throttling back the clock, and jumping execution between cores to keep any one core from overheating – all techniques that the chip will do automatically when configured properly. However, because all of those things slow down your throughput, it's important when performance testing your software to remember that you can't rely on full-speed execution. Spare processing capacity needs to be reserved to ensure the CPU doesn't fry itself or anything near it. If you're lucky enough to be building a plugged-in product without noise-volume restrictions, you can always employ fans for more efficient active cooling.

Excess heat is only part of the problem with a revved-up chip; it also consumes more electricity. Again, if you can rely on constant electrical power from the grid, this is a secondary concern. However, for battery-driven products this can be a serious worry. For a battery-dependent design, you're probably going to want to go with an i.MX 6ULL, which sips current compared to its big brother, the i.MX 8. The i.MX 8M Mini is the most power-sensitive chip in the i.MX 8 lineup, and it's still running at twice the bus width, three times the clock speed, and with three more cores than the diminutive i.MX 6ULL.

#### 5. Support chips

One of the really intriguing aspects of the i.MX 8 family is that alongside the Cortex-A35/A53/A72 that runs the OS and standard tasks is one (or more) Cortex-M8 cores. The Cortex-M is Arm's microcontroller profile. A Cortex-M architecture has no memory management unit (so, no memory-managed OS), no cache controller (no high-speed operations), and no floating-point unit (slower number crunching).

## NXP has added safety critical support with the i.MX 8X in automotive and other functional safety applications.



The Cortex-M can update safety-critical portions of a display without requiring Cortex-A coordination.

However, it runs on very low power and is real-time responsive, making it ideal for dedicated real-time activities like reading or reacting to sensors, controlling hardware peripherals, maintaining a system watchdog, or managing battery and low-power states.

Many systems use a separate independent microcontroller to manage these tasks – with an i.MX 8 you can build it into a single chip, potentially saving cost and shrinking board space. That in itself is a great reason to consider a move to an i.MX 8 design.

### 6. Safety critical

Another intriguing add-on to the i.MX 8 lineup is their support for safety critical applications. With automotive projects increasingly needing both safety critical and non-critical parts, the i.MX 8X seems tailor-made to win automotive hearts and minds, as well as any other applications that need a mix of horsepower with functional safety. Not only does the i.MX 8X provide support for error-corrected memory, but the Cortex-M core can independently access the graphical stack. That lets the Cortex-M run safety-critical processes that can run outside of the main application space yet still provide guaranteed display of critical diagnostics, warning tell-tales, or any other mission-critical information. Certification of code running on the Cortex-M should also be easier with a simpler task model and deterministic execution behavior.

Even once you've narrowed down your choice to the i.MX 8, there are many variants to choose from.

The i.MX 8M Mini  
provides  
performance  
with low power  
consumption  
at budget prices.

## Variants

If the i.MX 8 series still seems appropriate for your project, you're probably wondering which variant makes the most sense. We can't cover every consideration in this paper, but here are the main questions for deciding on the i.MX 8 family.

### **Q: Do you need functional safety?**

Yes – i.MX 8X family: i.MX 8QuadXPlus, i.MX 8DualXPlus, i.MX 8DualX

### **Q: Do you need 4K video or maximum performance?**

Yes – i.MX 8 family: i.MX 8QuadMax, i.MX 8QuadPlus, i.MX 8DualMax

### **Q: Do you need decent performance but a “standard size” user interface?**

Yes – i.MX 8M Nano family: i.MX 8M Nano Quad, i.MX 8M Nano QuadLite, i.MX 8M Nano Dual, i.MX 8M Nano DualLite, i.MX 8M Nano Solo, i.MX 8M Nano SoloLite

(Note that contrary to the expectation you might have from the Nano naming convention, this series has more power than the Mini series.)

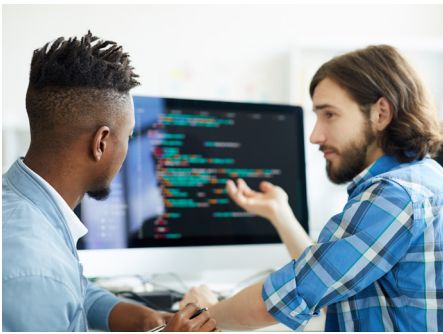
### **Q: Do you need performance with low power consumption or a budget price?**

Yes – i.MX 8M Mini family: i.MX 8M Mini Quad, i.MX 8M Mini QuadLite, i.MX 8M Mini Dual, i.MX 8M Mini DualLite, i.MX 8M Mini Solo, i.MX 8M Mini SoloLite

## There are many factors to take into account in order to make the right hardware choice.

### Summary

Whether you are considering an NXP i.MX 8, NXP i.MX 6, or another vendor's silicon, there are many factors to take into account in order to make the right hardware choice. If you'd like some of our advice in narrowing down your silicon decisions, we'd be happy to help.



We can help you decide on the right hardware to build your products.

### About the KDAB Group

The KDAB Group is the world's leading software consultancy for architecture, development and design of Qt, C++ and OpenGL applications across desktop, embedded and mobile platforms. KDAB is the biggest independent contributor to Qt and is the world's first ISO 9001 certified Qt consulting and development company. Our experts build run-times, mix native and web technologies, solve hardware stack performance issues and porting problems for hundreds of customers, many among the Fortune 500. KDAB's tools and extensive experience in creating, debugging, profiling and porting complex applications help developers worldwide to deliver successful projects. KDAB's trainers, all full-time developers, provide market leading, hands-on, training for Qt, OpenGL and modern C++ in multiple languages.



[www.kdab.com](http://www.kdab.com)