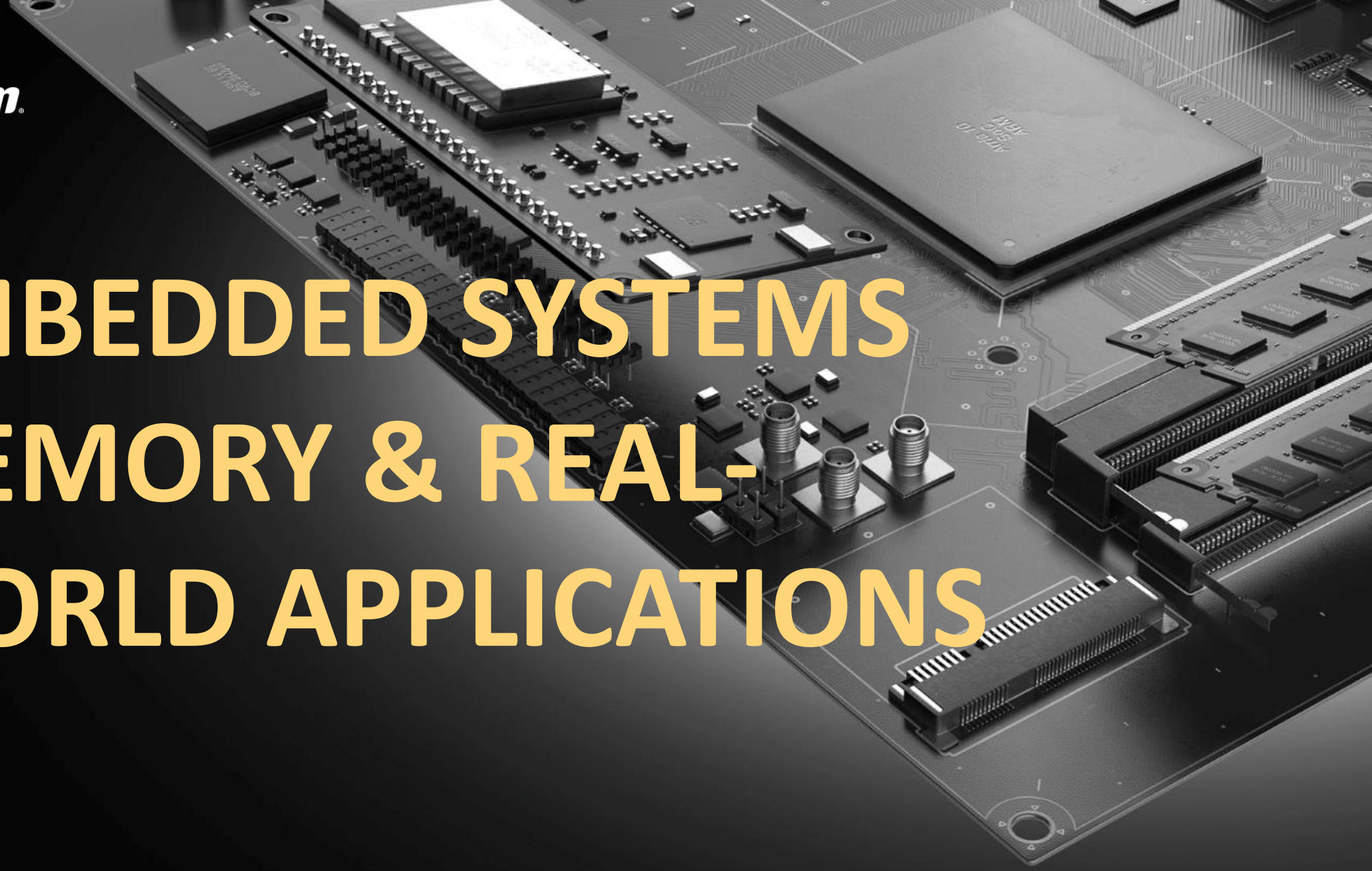
Altium

EMBEDDED SYSTEMS MEMORY & REAL-WORLD APPLICATIONS



Altium

1. LOW POWER MEMORY MANAGEMENT FOR EMBEDDED SYSTEMS
2. HOW TO PERFORM MEMORY TESTING IN EMBEDDED SYSTEMS
3. HOW FRAM MEMORY SIMPLIFIES EMBEDDED SYSTEM DATA LOGGING
4. HOW EMBEDDED MACHINE LEARNING APPLICATIONS WILL BENEFIT FROM 5G AND THE CLOUD
5. EMBEDDED ACTIVE COMPONENTS ARE ENABLING INDEPENDENT SMARTWATCHES WITH SMARTPHONE CAPABILITIES
6. MECHANICAL DESIGN AND 3D MODELING ASSISTS WITH PCB DESIGN IN EMBEDDED SYSTEMS

Altium

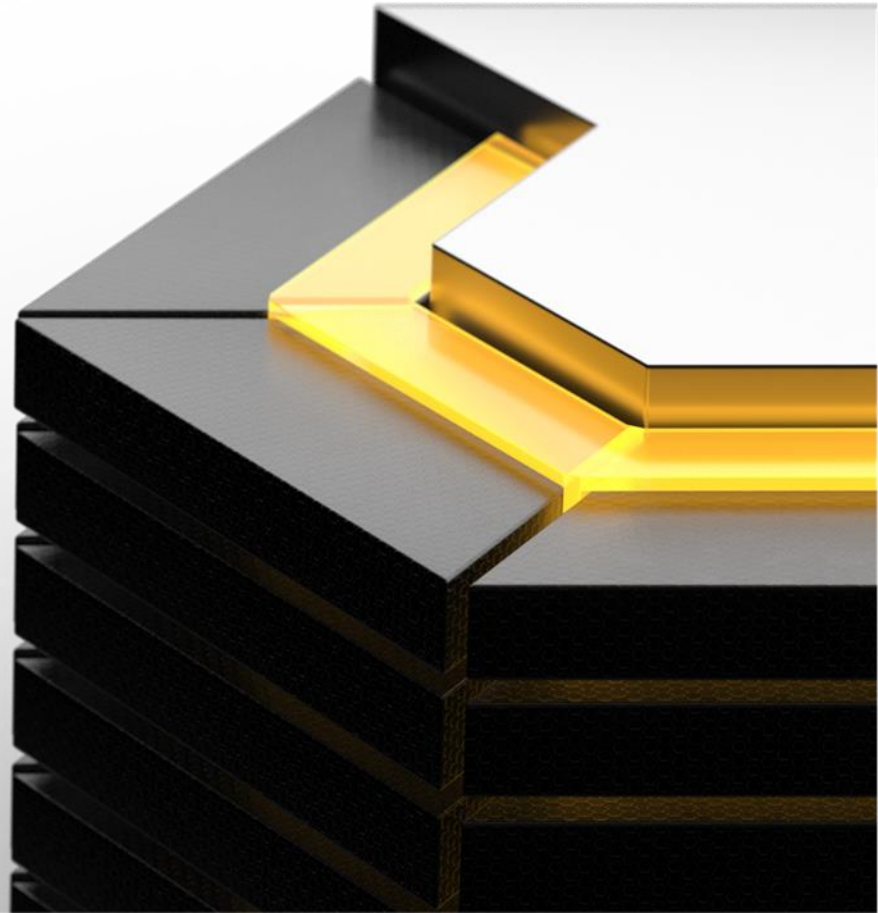
1. LOW POWER MEMORY MANAGEMENT FOR EMBEDDED SYSTEMS

Altium

Access Costs in Low Power Memory Management

In embedded systems, we're often concerned about speed, especially when your software is running a vehicle. However, we're also worried about energy usage. We all know that computations can burn through energy, but memory operations also have a significant effect. That effect gets compounded the further away from the processor cores you get.

- The two main memory actions we deal with are data accesses and instruction fetches. It's estimated that data access accounts for 40% of memory processes, while instruction fetches make up 60%.
- It's important to think about which operations you're optimizing for when designing your program.
- This estimation would suggest that you should spend your time trying to fetch instructions more efficiently, rather than worry about data access.

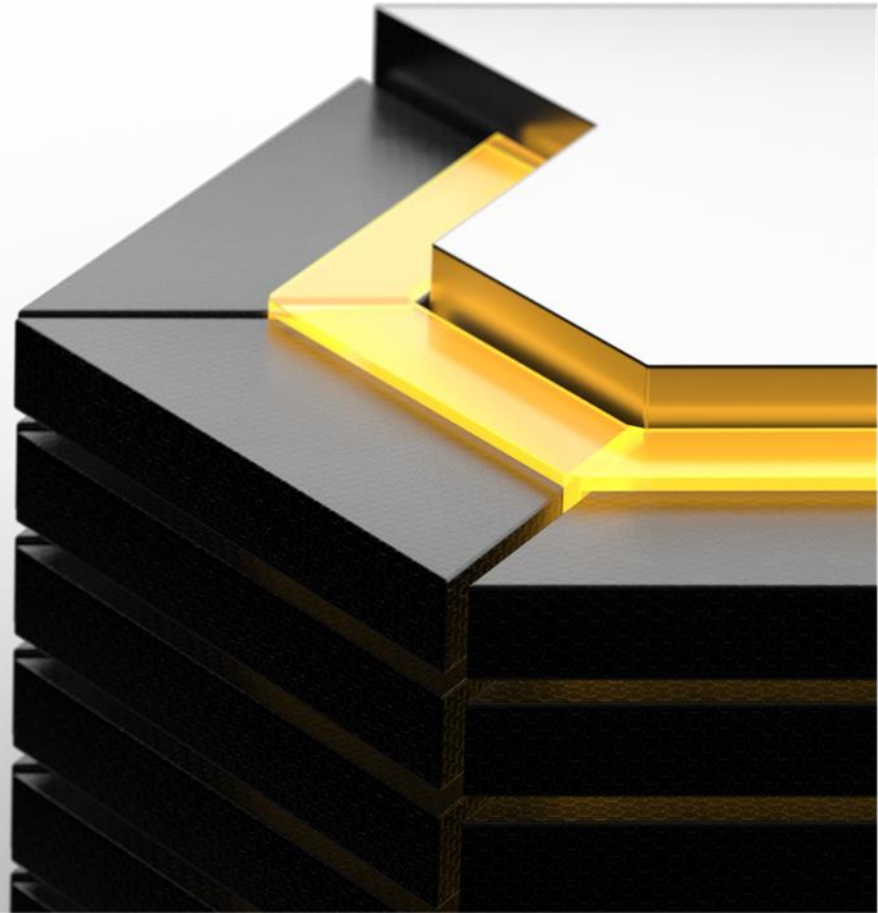


Memory Type and Energy Usage

There are many types of memory, but the two we'll be comparing are SPM and cache memory. Scratchpad memory is generally fast and wide SRAM with a very direct connection to the core. SRAM costs a good deal more than DRAM, and at some point, the monetary cost will outweigh the energy savings. However, cache memory has several advantages over SPM, including:

- Ease of use
- Availability
- Caches are more easy to access and don't require you to manage them as much as SPM.
- They'll automatically adapt to your software while it's running without your intervention
- While SPM and cache memory are both great, you should be sure not to mix and match them.

[See original content here]



Altium

1. LOW POWER MEMORY MANAGEMENT FOR EMBEDDED SYSTEMS
2. HOW TO PERFORM MEMORY TESTING IN EMBEDDED SYSTEMS
3. HOW FRAM MEMORY SIMPLIFIES EMBEDDED SYSTEM DATA LOGGING
4. HOW EMBEDDED MACHINE LEARNING APPLICATIONS WILL BENEFIT FROM 5G AND THE CLOUD
5. EMBEDDED ACTIVE COMPONENTS ARE ENABLING INDEPENDENT SMARTWATCHES WITH SMARTPHONE CAPABILITIES
6. MECHANICAL DESIGN AND 3D MODELING ASSISTS WITH PCB DESIGN IN EMBEDDED SYSTEMS

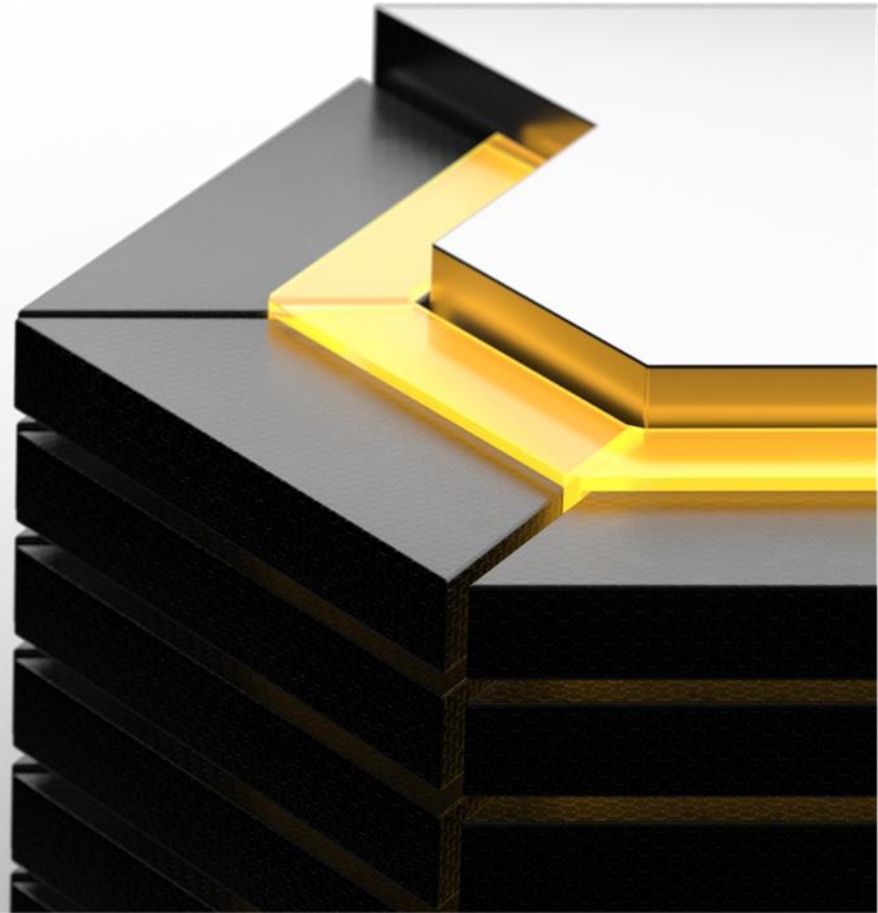
Altium

2. HOW TO PERFORM MEMORY TESTING IN EMBEDDED SYSTEMS

The Importance of Testing

As an engineer you know that an untested system is an unreliable system. When it comes to cars with advanced driver assistance systems (ADAS), instability is unacceptable. When it comes to storage, a memory protection unit (MPU) can help prevent cascading failures, but memory issues can still cause individual systems to fail. In-house testing can help ensure that your code doesn't cause any memory complications. As well, self-testing in the field will verify that your ADAS enabled vehicle is working properly before it starts driving.

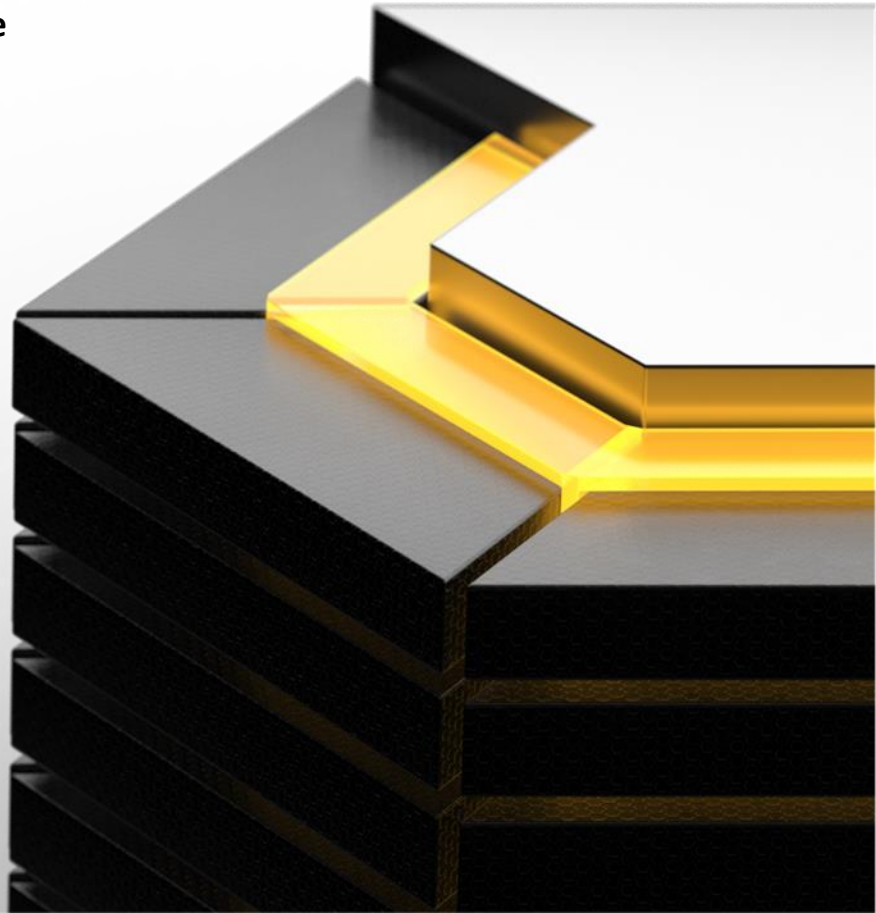
- The only way to perform checks during the development process and testing before deployment is with the right tools.
- A standalone debugger can speed up your work, and a static analyzer can help you check if your MPU is working properly.
- Using self-testing systems to check and see if your memory is working in the field will help you nip problems in the bud and reduce the risk of failure.



Common Memory Problems and Solutions

System storage has several different failure modes, some of which are not preventable. The ones we can deal with and will look at are memory leaks, memory fragmentation, stuck bits, and crosstalk.

- **Memory Leaks:** A memory leak occurs when your program doesn't free up memory that is used. It's important to use a memory leak detection tool before deployment. These tools can help you find issues before they actually manifest and cause havoc.
- **Memory Fragmentation:** Fragmentation occurs as your program carves out blocks of memory to use. You can get a memory mapping program that shows you memory usage as your system runs. Then you can make changes in your code to reduce fragmentation to an acceptable level.
- **Stuck Bits:** Sometimes specific bits in memory modules can get stuck to a 1 or 0. On startup, you can simply tell your program to write all bits of your memory to 1 and then to 0, then check them.
- **Crosstalk:** This kind of crosstalk occurs when changing one bit changes another unintentionally. To test for this you can write all 0's to your memory (like in the previous test) then change each bit to 1 and back to 0 individually.



[See original content here]

Altium

1. LOW POWER MEMORY MANAGEMENT FOR EMBEDDED SYSTEMS
2. HOW TO PERFORM MEMORY TESTING IN EMBEDDED SYSTEMS
3. HOW FRAM MEMORY SIMPLIFIES EMBEDDED SYSTEM DATA LOGGING
4. HOW EMBEDDED MACHINE LEARNING APPLICATIONS WILL BENEFIT FROM 5G AND THE CLOUD
5. EMBEDDED ACTIVE COMPONENTS ARE ENABLING INDEPENDENT SMARTWATCHES WITH SMARTPHONE CAPABILITIES
6. MECHANICAL DESIGN AND 3D MODELING ASSISTS WITH PCB DESIGN IN EMBEDDED SYSTEMS

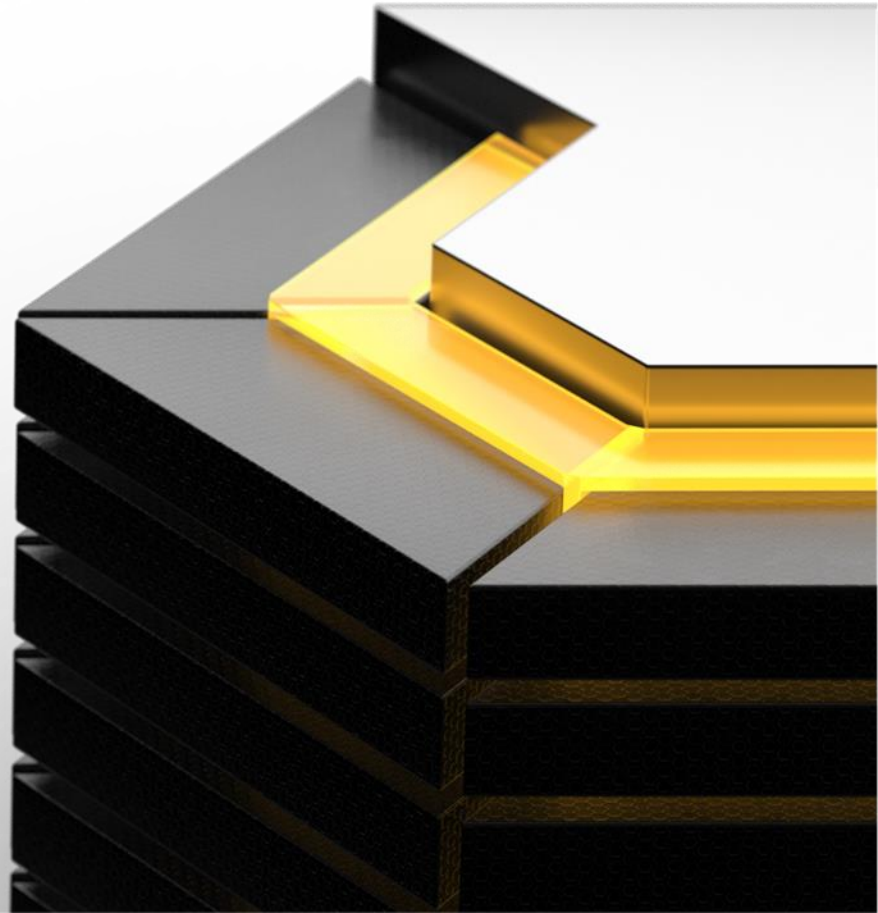
Altium

3. HOW FRAM MEMORY SIMPLIFIES EMBEDDED SYSTEM DATA LOGGING

HOW FRAM COMPARES TO SRAM AND FLASH

SRAM and Flash are both popular memory chips in their own right.

- SRAM is a form of volatile memory. This means that data stored in the memory is erased when the power supply is removed or cut-off.
- The great thing about SRAM is that it has an unlimited write-cycle, meaning it does not physically deteriorate as it is used.
- Flash memory is a non-volatile memory. This makes it particularly useful for storing transaction logs that must remain intact, even when the power supply is removed.
- The only downside to Flash memory is the low write endurance, which is often in the tens of thousands.
- FRAM is a memory chip that inherits the advantages of both SRAM and Flash memories. FRAM is a non-volatile memory that has an extremely high write endurance.
- The cost of FRAM has dramatically decreased as the manufacturing process of FRAM has matured.

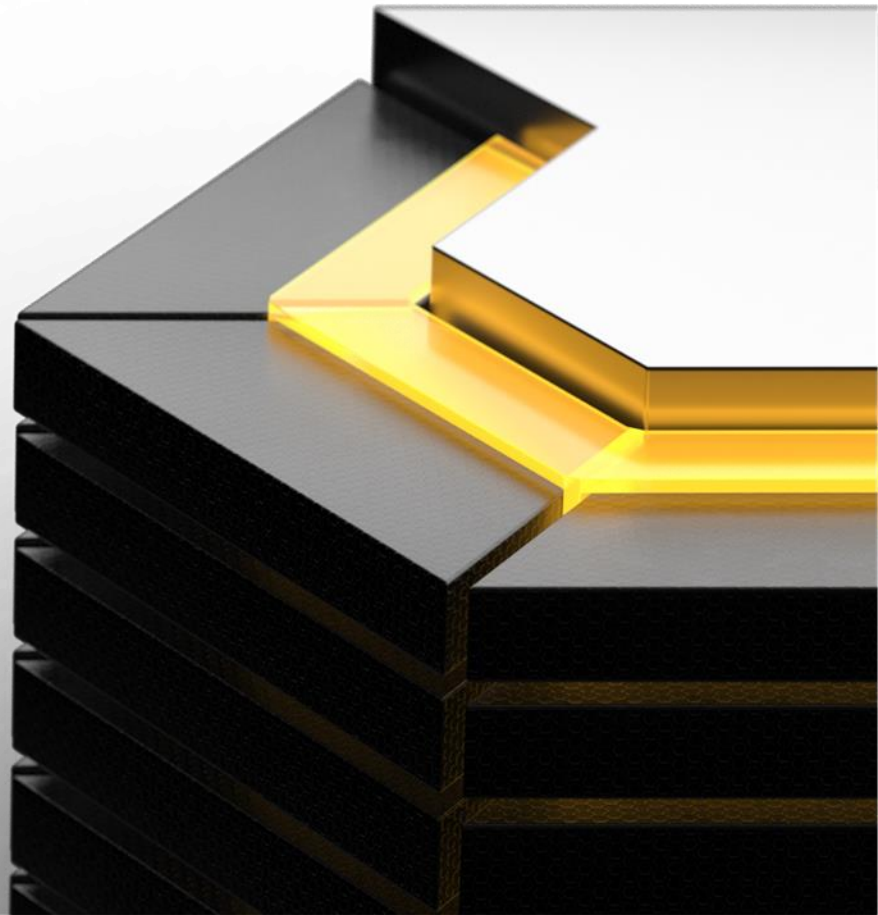


Datalogging with FRAM

Reliable data logging capability in embedded systems has always been mandatory regardless its application. FRAM changes how datalogging is designed in embedded systems.

- In embedded systems, tens of thousands records are stored in the memory, and they are constantly being retrieved by software applications to an external database.
- That means the microcontroller (MCU) itself has to be intelligent enough to keep track on the newest and oldest data and mitigate what happens when the memory is full.
- When FRAM became a commercially viable electronics component in the late 2000s, data logging became ridiculously simple.
- Data pointers could be saved in FRAM without the fear of power loss corruption or physical deterioration to the memory chip.

[See original content here]



Altium

1. LOW POWER MEMORY MANAGEMENT FOR EMBEDDED SYSTEMS
2. HOW TO PERFORM MEMORY TESTING IN EMBEDDED SYSTEMS
3. HOW FRAM MEMORY SIMPLIFIES EMBEDDED SYSTEM DATA LOGGING
4. HOW EMBEDDED MACHINE LEARNING APPLICATIONS WILL BENEFIT FROM 5G AND THE CLOUD
5. EMBEDDED ACTIVE COMPONENTS ARE ENABLING INDEPENDENT SMARTWATCHES WITH SMARTPHONE CAPABILITIES
6. MECHANICAL DESIGN AND 3D MODELING ASSISTS WITH PCB DESIGN IN EMBEDDED SYSTEMS

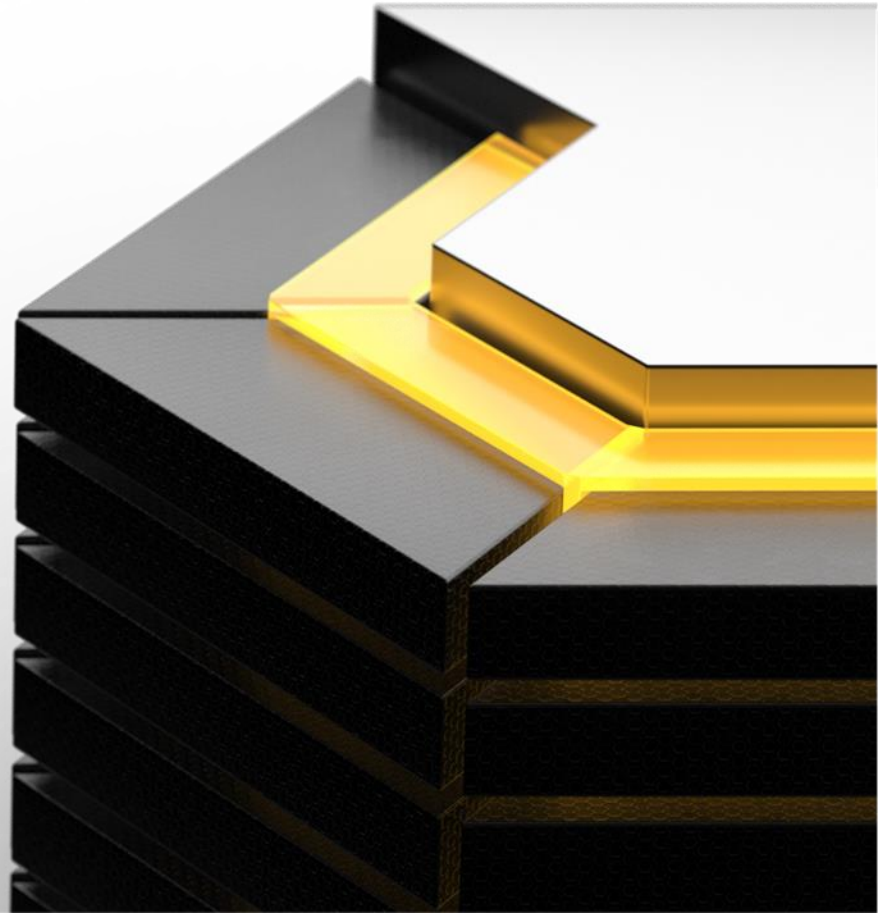
Altium

4. HOW EMBEDDED MACHINE LEARNING APPLICATIONS WILL BENEFIT FROM 5G AND THE CLOUD

Machine Learning in Embedded Systems

Machine learning is not a new concept, but continuing advances in processing power are making it a reality. AI will allow gadgets to interact with their environments much more intelligently.

- The advantages of intelligent sensing are huge for embedded systems. They're already being put into use by some devices.
- Machine learning takes a humongous amount of processing power, more than is prudent to put in an embedded system.
- Developers are now using GPUs to speed up machine learning processes, but those won't get chips down to the size you need.
- A shorter term solution for bringing AI to embedded systems is 5G and cloud computing.

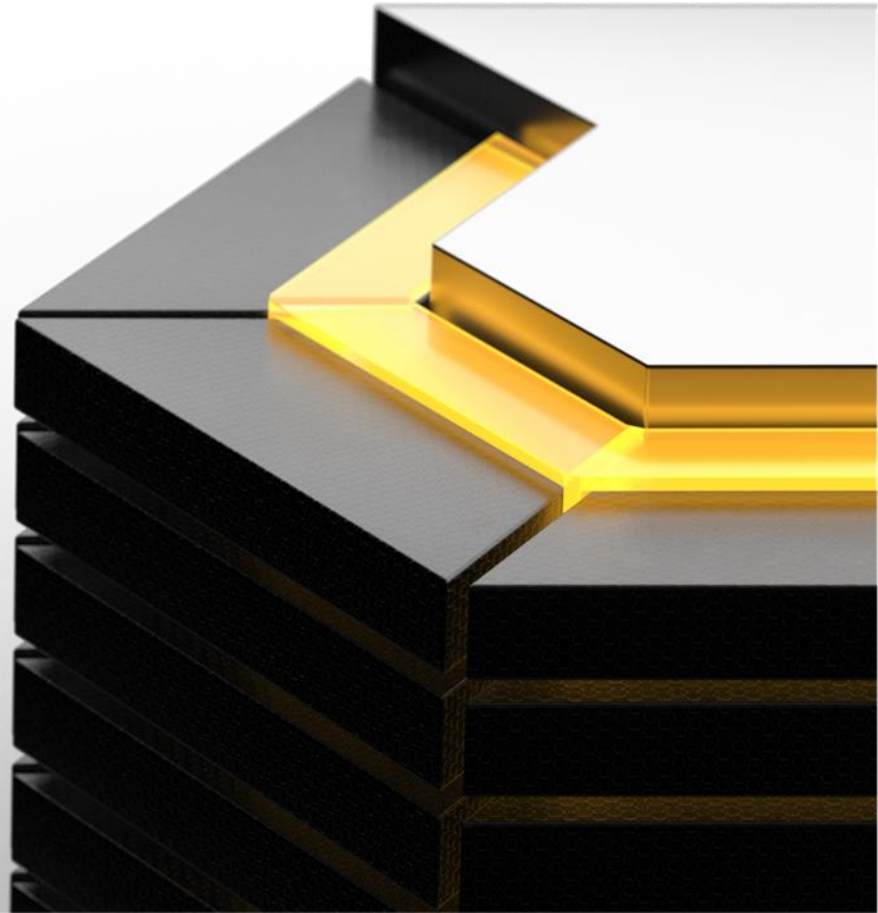


Cloud Computer and 5G

Cloud computing can solve the aforementioned power and processing limitations. Calculations can be run at an external location with effectively infinite computing power and a grid connection.

- People are already using the Cloud for distributed computing, so there's no reason that embedded systems shouldn't do the same.
- Intel estimates that ADAS enabled cars will need to process 1 GB of data per second. That's a lot of information to be transmitted over a WiFi or 4G connection.
- 5G will support data rates up to 10 Gbps with latencies under 10 ms.
- This means that your device will be able to transfer the required data and almost instantly receive an interpretation.
- Companies are also working on low power 5G antennas so that your boards can make smarter decisions using less electricity.

[See original content here]



Altium

1. LOW POWER MEMORY MANAGEMENT FOR EMBEDDED SYSTEMS
2. HOW TO PERFORM MEMORY TESTING IN EMBEDDED SYSTEMS
3. HOW FRAM MEMORY SIMPLIFIES EMBEDDED SYSTEM DATA LOGGING
4. HOW EMBEDDED MACHINE LEARNING APPLICATIONS WILL BENEFIT FROM 5G AND THE CLOUD
5. EMBEDDED ACTIVE COMPONENTS ARE ENABLING INDEPENDENT SMARTWATCHES WITH SMARTPHONE CAPABILITIES
6. MECHANICAL DESIGN AND 3D MODELING ASSISTS WITH PCB DESIGN IN EMBEDDED SYSTEMS

Altium

5. EMBEDDED ACTIVE COMPONENTS ARE ENABLING INDEPENDENT SMARTWATCHES WITH SMARTPHONE CAPABILITIES

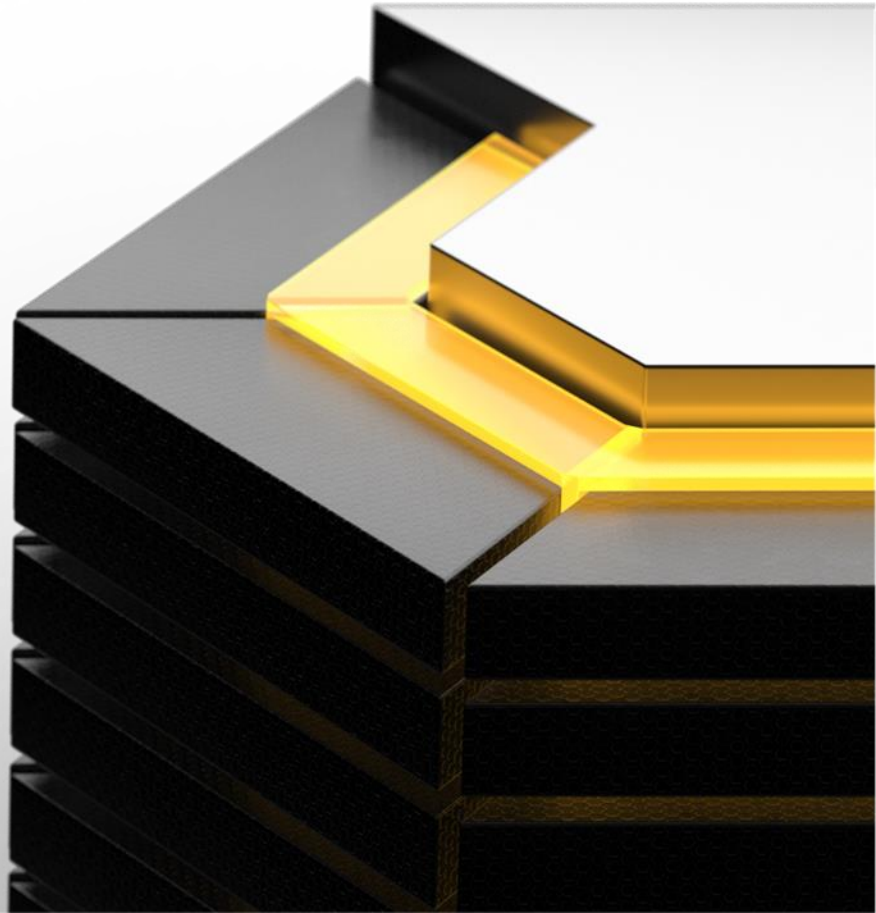
Altium

Current State of Smartwatch Tech

Current state of smartwatch technology fails to deliver broad use cases. Why not? Because currently they don't do anything that something else can't do better.

- Only 0.1% of the world's population bought one in Q4 2016.
- According to a 2016 survey of 11,000 consumers by Kantar Worldpanel ComTech, the strongest objection of 33% of consumers to owning a smartwatch was that smartwatch functions weren't useful to them.
- New technologies have the potential to integrate watches into everyday life in ways phones have yet to fully realize.

See original content here



Altium

1. LOW POWER MEMORY MANAGEMENT FOR EMBEDDED SYSTEMS
2. HOW TO PERFORM MEMORY TESTING IN EMBEDDED SYSTEMS
3. HOW FRAM MEMORY SIMPLIFIES EMBEDDED SYSTEM DATA LOGGING
4. HOW EMBEDDED MACHINE LEARNING APPLICATIONS WILL BENEFIT FROM 5G AND THE CLOUD
5. EMBEDDED ACTIVE COMPONENTS ARE ENABLING INDEPENDENT SMARTWATCHES WITH SMARTPHONE CAPABILITIES
6. MECHANICAL DESIGN AND 3D MODELING ASSISTS WITH PCB DESIGN IN EMBEDDED SYSTEMS

Altium

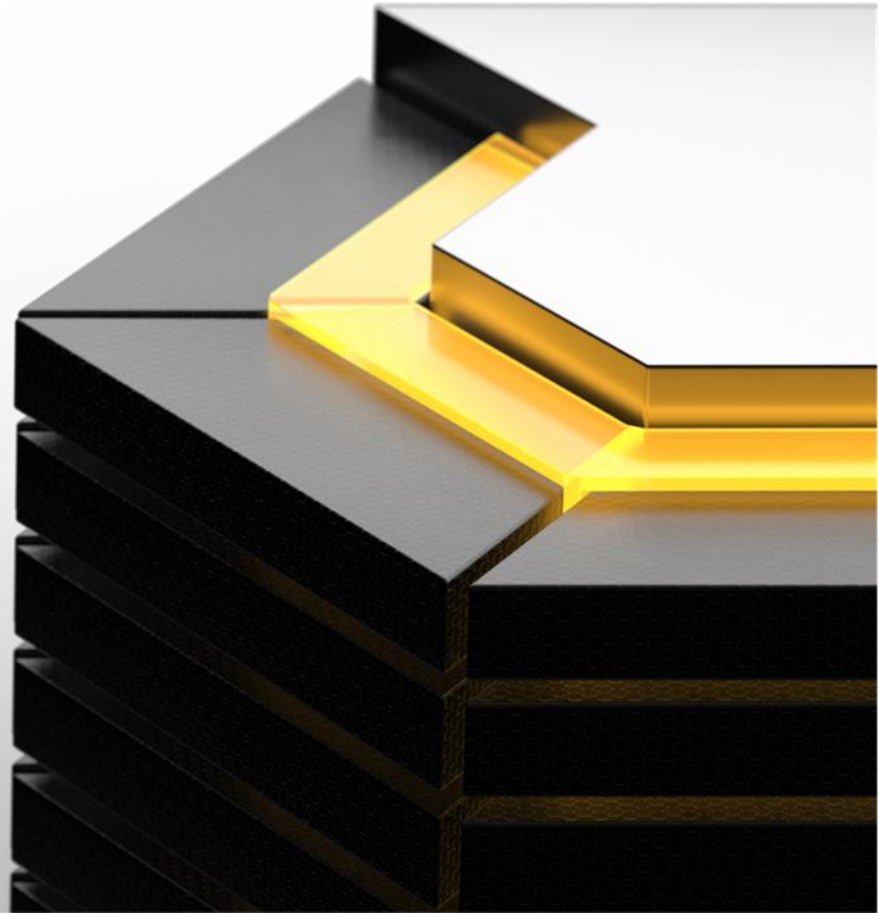
6. MECHANICAL DESIGN AND 3D MODELING ASSISTS WITH PCB DESIGN IN EMBEDDED SYSTEMS

Electrical Design for Mechanical Applications

Unlike mechanical and electrical engineers, the IoT and ADAS have found common ground in embedded systems. For both applications, PCBs need to be designed with mechanical limitations in mind. That means you'll have to be the bigger person and enhance your design process with some mechanical techniques. Process enhancement looks like two things: collaborating more with lesser engineers, and appropriating their design tools.

- Collaborating more with MEs at the beginning of your design process will mean less ECOs later, and a board that is better suited for its mechanical environments.
- Learning 3D modeling makes sure you know what kind of mechanical factors your board will be exposed to. New cutting-edge PCB design software can help you by automatically creating a 3D model for you.
- PCB designers will have to master 3D modeling if they want to take full advantage of future 3D design capabilities. If you don't start now, you may find mechanical engineers moving into future PCB design, just because they have the 3D design skills.

[See original content here]



Altium

Thanks for your attention!