Technical Challenge – The Problem

- There are two cores on a device with different CPU architectures.
- The first core's (CPU1) memory is 8-bit addressable, i.e. each address in memory references 8-bits of data.
- The second core's (CPU2) memory is 16-bit addressable, i.e. each address in memory references 16-bits of data.
- The cores share some limited block of memory. This memory is 512-bits. Both CPUs can read from this block of memory at once, however concurrent writes are **not** allowed.
- Design and implement a module which allows CPU1 to pass an array of characters of unspecified size for CPU2 to consume.
- Design a complementary module for CPU2 to receive and correctly parse this data, before passing it to some user.
- Additionally, write unit tests using the framework of your choice to test your new APIs.
- Finally, prepare a short (5-10 minute) presentation on your chosen design and implementation.

Technical Challenge – Requirements

- Application shall be written in C11 (no C++) with no additional libraries.
- CPU1 shall have a single API which has an array of characters as input.
- CPU2 shall have a single API which can return the received data to the user through some mechanism.
- CPU1 shall receive some indication from CPU2 once it has received new data.
- The input array of characters may exceed 512-bits.
- The shared memory block shall live at address 0x20000000-0x20000040 on CPU1.
- The shared memory block shall live at address 0x80010-0x80030 on CPU2.
- Solution shall be on a publicly accessible repository on GitHub.