Twitter Thread by dustin

dustin
@32mbit

rubber duck thread Using the Sega CD subcpu in mode 1 is very doable but the larger problem is: what kind of tasks can we hand off to it, and are they asynchronous?

We can't run code directly off the cart from the subcpu. We can't simply make the faster cpu run the game. So it makes more sense that we'd have subroutines running on the subcpu to accelerate common operations that can be called from the MD main cpu.

But we run into a problem here. First is defining what "common operations" would really be. Since the subcpu is another 68k with a roughly 50% higher clock, it wouldn't make much sense to run very small or atomic operations on it. There is overhead in passing arguments...

... because it has to be done in shared memory. Overhead could kill the gains. We'd get more benefit from a larger, more complex function to really take advantage of that higher clock speed. This adds another layer to the problem, synchronicity.

If the MD cpu has to block (stop and wait for a result from the subcpu) we're losing a lot of potential throughput. The effect is similar the other way around in that we also waste resources on the subcpu waiting for a command from the MD.

(BTW this is why using the dual SH2s on the Saturn wasn't super popular, as it requires a similar semaphore setup. The game would have to be carefully crafted to utilize it at the planning stages.)

If you can set up a task that runs asynchronously, that is, both CPUs working away, then that's ideal. It's just that for a lot of games, this isn't really feasible or useful. Most game loops on this old hardware are essentially single threaded and each step is done in order...

...every time, for every frame, and if there isn't time to do it all? Slowdowns. This isn't true of every game of the era, but it's true the vast majority of the time. So ultimately, this makes the entire approach not attractive for rom hacks. Much like a Saturn game...

...you'd have to arrange the game loop as such to not waste time and resources. To do certain things asynchronously. Nearly impossible for a rom hack. But what about homebrew and new games going forward? There is a bit of promise here

but still some caveats.

The best reason to use the subcpu is to get access to the cd hardware features: the disc drive, the PCM, the additional RAM, the ASIC graphics scaling features. Using it to accelerate CPU bound operations is probably not worthwhile.

What if you designed a game to use resources carefully, though? You're still a bit hampered by the fact that the two CPUs can't run the same code out of the same memory, so no clever threading tricks. Also at this point you may as well be making a a Sega CD game!

To this end I'm experimenting with creating a small library, intended for homebrew, but similar to MSU-MD, that will allow the user to play PCM, CDDA, and (hopefully!) use the scaler. I don't know if this will mature and it depends on how useful it ultimately is.