

The not-so-mega MEGA65?

Sometimes I see people posting their unsolicited opinions about the coming MEGA65 on social media. The talk is that this is the heir to the C64 but, it is just another emulator of a prototype machine. “We already have the Ultimate-series and the MISTer, we don’t need another one. It’s too expensive for what it is.” In this article I will try to address that opinion, but I will also share my initial experience with the machine and try to give you an in-depth impression of its capabilities through the eyes of a demo scener.

Before I go any further, allow me to introduce myself. I have been a part of the C64 scene since the mid-80s. I never really played games. Shortly after I got my first C64 I discovered the wonder of demos. I found the crack intros more appealing than the actual games. So, that ended up being my trade. Demos. I had some years of inactivity in the 90s and 00s. Got back into the scene in the early 10s and am still quite active in the C64 demo scene today.

So, what IS the MEGA65? Is all the hype, correct? Is this just another emulator? How is this machine even relevant with an already overpopulated retro computing market? Even if it can find a niche of its own, why would this machine be interesting to me?

Well, let’s try to find out. I will try to explain what the MEGA65 is from MY point of view. This is by no means a sales article or a critique of the project. It is just my opinions. Use them or lose them. So, let’s get cracking.

Pi in a box and the elephant in the room

“Pi in a box”, somebody wrote on social media. Probably with the Mini and Maxi machines in mind. Whoever wrote that clearly has absolutely no insight into what this really machine is and what the team behind it are trying to achieve. Before I go any further, let’s all agree on what a field-programmable gate array (FPGA) is. It is NOT an emulator. It is NOT a piece of software. It is NOT off-the-shelf hardware installed in a fancy chassis. What it IS though, is a completely blank programmable chip in a box delivered on a custom-made motherboard. Using VHDL this blank chip can be programmed to work as any other set of chips by setting and clearing gates – just like any other microchip. Emulators, on-the-other-hand, have software trying to copy specific hardware behaviours. That is a completely different beast. So, saying that the MEGA65, which happens to be FPGA, is just a “Pi in a box” and thus insinuating that it is “just an emulation” is plain wrong.

That brings me on to the next and much more important thing about this machine: The MEGA65 is the MEGA65. Now, please read that sentence a few times and let it sink in. Yes, the public thinks of the MEGA65 as a realisation or maybe even a replication of the original Commodore 65. Okay, here comes The Important Stuff. The MEGA65 is indeed based on the C65. It is presumably fully capable of doing the same things that the C65s could. However, keep in mind that the C65 was never finished. Would you want an incomplete machine? No. So, the team decided that the MEGA65 should be two things:

1. The completion of the Commodore 65 as the MEGA65 team imagine it could’ve looked like

2. The Commodore 65 but implemented as the MEGA65 team can best do today

What does this mean? Well, it means that the c65s are prototypes, they are just that, prototypes – by no means finalized machines. There is no right or wrong in the implementation of a successor. You could just run the MEGA65 in Commodore 65 mode and congratulations! With the right settings, a Commodore C65 with all the bugs and no software to run on it.

I think the team did the right thing. They are re-working the entire ROM of the C65, fixing bugs, making improvements, and basically completing an incomplete design. Add to that, they are also making an Open ROM, which means an implementation without any licensing issues. Okay, so that gives us a working Commodore 65. However, when you start looking into the hardware specifications and the design of that prototype, you will realize that mistakes were made. I will get into details about that later, but for now, let's just say the Commodore 65 design was flawed in some ways and the MEGA65 team filled in the holes.

Point 2. Being that this is also a dream machine, when you first start the machine, you will get something that looks very similar to the start screen of the C65, but there is SO MUCH MORE under the hood. This is where things start to become interesting. If I told you that you could have the familiar environment of the C64 as you know it but add loads and loads of more power and features into that platform. THAT'S the MEGA65. Read on and I'll try to peel off some layers, so you get an understanding of what I mean.

Hardware

The MEGA65 is a dream machine based on the Commodore 65. I've said before and I'll say it again. That is how I see this. The MEGA65 allows you to run in "native" C65 mode, 3.5mhz. As a "bonus" the MEGA65 can also run at 40Mhz. I have trouble finding any argument why this isn't exciting.

The MEGA65 has 384kb of memory. That obviously presents some challenges in terms of addressing all this memory with an 8-bit processor. The MEGA65 works bit like the Amiga, you have "chip" memory (accessible to the VIC chip) and you have "slow" memory. You get the idea. The C65 uses banking, something that is also used on the C128. It may make it harder to learn than on the C64. It was and still is for me, I am still learning. However, with a little bit of persistence you get far.

The original design of the Commodore 65 has one major flaw that pretty much makes it completely irrelevant now and back then, when it was supposed to be the poor man's Amiga. The problem here is the VIC-III chip. This chip does graphics in the way Amiga's do, meaning you have multiple bit planes that you stack. The more colours you want, the more bit planes you stack.

So, imagine having the 4510 (the C65 CPU) running at 3.5Mhz trying to update not one, but *SEVERAL bit planes*. The 6502 has poor bit-wise processing support! This is not going to end well. I am sure that if the machine had been released, developers would've found ways to make it work, but this is really a flawed design in my opinion.

So, you have full access to the VIC-3 chip. Yes, that means an FPGA implementation of the chip – NOT an emulation, remember? That's not cool, so the team made a genius decision. They took the original VIC-II

design from the C64 as we know it and made changes so that it would become THEIR dream VIC chip, the VIC-4. This is not the graphics chip that Commodore made. This is the MEGA65 team going crazy. They made a chip that is ridiculously powerful, and it shares a lot of the properties with the original VIC-2. They added new graphics modes and new features.

So, let's talk about that for a minute. I won't cover everything, but just a few of the things that I found cool... and actually... also a bit worrying. I'll get back to that. The VIC-4 has a few extra registers in the \$Dxxx area, so nothing scary here. You can now do stuff like expanding the border with just a few pokes. No more difficult coding to ensure stable rasters because you can easily ensure stability running at 40MHz with a high resolution raster register – if you should ever need it – because that's how fast the machine is. What I try to say is that a simple

LDA \$d012

wait: CMP \$d021

BEQ wait

will almost get you there, however all the issues that required us to make stable rasters on the c64 are gone on the MEGA65. No borders, no need to split rasters – there are just other ways to do these things.

You still have \$D020, \$D021 for screen colours. You have a different register for X and Y smooth scrolling, but instead of only moving 8 pixels, you can now move 256. You have new properties for your sprites, give them more colours, make them 128pixels wide (x-expanded 64bit sprites) and even make them fill the entire screen by enabling X-position raster so you can multiplex sprites horizontally. However, you don't really want to do that. Sprites in this machine seem weirdly redundant. Why? The RRB. The Raster Rewrite Buffer. What is that? It is a time machine.

As you know in the C64 when making raster programming, what is done is done. However, the RRB allows you to go back in the raster line the machine has just drawn and tell it to do some more. Like, hey I've drawn a cool picture but let's go back to pixel position 20 and add a layer of text, then go back again and add another layer of something and again and again.

The RRB allows you to make an incredible number of layers. THAT's why sprites may not be so relevant anymore. The best part? The RRB takes NO processor cycles. The only processing you do, is to update the tables that tell the RRB what to do. How does that work? Before I explain that, I must present another new feature: 16bit charset. Yeah, 8192 chars in a charset. Not 16bit totally, I know but there's an explanation I won't cover here. Anyway, with 8192 unique characters you can fill the entire screen, so ... why would you need bitmap? Did I mention each character can have 64 unique colours out of a palette of 255? This is called Full Colour Mode (FCM).

The problem with FCM is that it does take quite a bit of memory, 8x8 bytes per char, so that's 64kb for a normal 40x25 char screen where each character is unique. Maybe not a problem since you have a healthy amount of memory. However, they added another mode, Nibble Colour Mode. This reduces the number of unique colours to 16 in a 16x8px square. I won't get into the tech details, but it is a 16x8 char mode

that allows 16 unique colours in such a field. Who needs 64 colours in 8x8 anyway? Remember, you still have 256 colours in the palette.

Speaking of colours. I'm still not sure what can be done, but there are 3 areas in memory, \$D1xx, \$D2xx and \$D3xx, for red, green, blue. So, index 0 is the RGB code of location \$D100, \$D200 and \$D300. So when you set screen colour, lda #0, sta \$D021, what you really do is you reference the palette lookup tables. Cool, even I can understand that.

If you want to relocate the screen area, you can now do this on byte level. Full screen scrolling is done using the usual smooth scroll registers (remember that also has some improvements) and then moving screen pointers – just like on the Amstrad in case you were wondering. So, making a MEGA65 version of Logon's Run will be easy. The same goes with the charset and the color ram.

At the time of writing this, the sound is problematic. I am 100% confident the team will get things sorted, but right now they have decent 6581 implementation and a non-working 8580 implementation. So, let's rewind back to the original specs of the C65. That stated 2 8580's. The team decided to do more and doubled that. So, you will have 4 8580's in the MEGA65. Remember this is FPGA so there will be updates and bugs will be fixed. Also, the MEGA65 allows you to play samples. There's a lot to be discovered but unlike the Amiga's 8-bit samples, the MEGA65 lets you do 16-bit and high sample rates. It also has a clever 4-bit mode that allows you to compress 2 samples into the same memory by simply splitting 8 bits into 2x4bits.

I have no idea where sound will go for the MEGA65. I know a lot of people are extremely passionate about the SID, but I also know there's a lot of potential in making Amiga-level music, which is very much possible on the MEGA65. Back to the fantasy machine, I don't think that would have been possible in the original C65 design. Getting the 8580 SID to work well is important I think because I think that will become the dominant choice for demo makers, but then again – that's just my opinion with my c64 upbringing. Amiga nerds may have a different opinion.

When I first started coding on the C65 I had to learn a new memory layout. We all know our C64, screen in \$0400, BASIC in \$A000, I/O in \$D000 and KERNAL in \$E000. The C65 has a slightly different layout, but that's just a matter of learning new habits. Screen is at \$0800, BASIC start \$2000 etc. Once you get comfortable with it, it's like riding a (C64) bike.

Let me just share a few of my favourite new processor features and instructions. Two come to mind. There's a lot more, but I am still learning. The C65 has a Z register. We finally get another register! My new favourite mnemonics must be

TAB and **TBA** set the base page or the zero page start address to anywhere in memory.

INW and **DEW** increase and decrease a word.

Branches now support 16-bit relative.

Somehow it feels like a forbidden pleasure because the machine is so incredibly fast that you tend to get a bit lazy in your optimizations.

Before I conclude my hardware coverage, let's talk about the blitter. Wait what? The Amiga blitter? Yeah, well not really. Let's just talk about the DMA that sits in the MEGA65 because it is similar. The DMA was a separate chip in the original C65 design, but they made it an integral part of the MEGA65 design I guess they just had to sneak in a few improvements. In short, DMA allows you to move massive amounts of data around in memory fast. For example, moving 8kb of memory takes a little less than one raster line. The DMA does that.

The best thing? When you start exploring the machine, you will realize that this is very much like the C65 you know, but just more... a LOT more. When I first sat down and started exploring, I found myself being that 13yo curious kid again, figuring things, and I still do that. Right now, I think THAT'S the best argument for buying a MEGA65. It will give you the comfort of home from the C64, but it will also broaden your horizon immensely.

Demo or die

Myself being a demo coder, my approach to the machine was: "What can I break"? What can I do that's seemingly difficult? After all, demo making is about bragging rights. Make the coolest routine. Make the highest number of dots. Make the real time vector phong shading. All that. That's the essence of the demo scene to me. Albeit I would argue that even on the dusty C64 the demo scene has evolved and storytelling, visual and flow design have entered the stage as dominant factors in making a quality demo. The MEGA65 will challenge us in new ways. I find myself on shaky ground here. The machine is still being developed and there are areas that I dare not touch yet, because they may change. I guess that's the price for buying a devkit, right? That's all cool. I am sure that when the final machine is released to the public, it will be with a promise of non-breaking updates looking ahead.

The MEGA65. Allow me to pause, no I will not talk more about the C65. Why? Because the MEGA65 is the logical superlative, the potent big brother that the C65 should have been. The C65 was a good PROTOTYPE. I am glad it never came to market because it would have failed. Judging from the prototype, alone. So, the MEGA65 will challenge the demo scene in new ways because suddenly all our known restrictions have been moved far into uncharted waters. There are still restrictions, but they are far, far away. Our challenge is to find these restrictions, but more importantly I think the demo scene on the MEGA65 cannot be like the C64 scene simply because the machine will allow us demo makers to include visuals and storytelling much more than we ever have before. I hope we won't see PC fly-by-forever demos which seems to be the consensus in that scene, but rather an evolution of the C64 scene. I guess when the C64 came out, it was pretty much the same and THAT'S the spirit of the MEGA65. Be that curious kid again. Be that guy who explores hardware. Only this time around, you'll have the luxury of a healthy head start.

Technically speaking you won't see raster splits and open border on this machine. I think you will see an increase in 3D demos (sigh) but the hardware also screams "abuse me" and I hope the majority of the old C64 scene will see that as I do. It will take some time for us to get there, but the journey is half the fun. So, compute like it's 1984 again, right?

The kids of yesterday need their drugs

The interesting question is if this machine even has relevancy in 2022? I am moving away from the core topic of this article but then again, I feel it's relevant to discuss. Most of the people I know in the retro scene are hitting 50 with the odd 30s guy. Is there a retro scene when we grow old or is this our last chance to relive our already lost youth? Well, lots of thoughts come to me but I think that the only way we can lose in this hobby and thus in our lives being these super nerdish people, is by not keeping on pushing ourselves in our passion for simple logic. I guess that's essentially what the old school demo scene is about anyway.

The MEGA65 is not a C65. Yes, it will run as a C65 and it will look like a C65, but it is a dream machine. It is a brand spanking new machine that lifts the original vision of the C65 to both completeness of the original design, but it is also a machine that ticks EVERY SINGLE BOX in the, yeah let's be 13yo again and explore. I personally love scavenging and scanning the manual, that is being updated almost daily, to find weird things and ways to bend the machine. It's a difficult beast to bend, but we will get there before we die!

Oh, did I really forget to mention that because of the nature of the design being FPGA-based, other cores can be implemented? The MEGA65 has a menu system that allows you to select a core to boot from. Right now it is being used extensively for testing different versions of the MEGA65 core, but in reality this could be ANY core. I know people in the scene are working on making smooth migration of cores to and from the MISTer platform is in the works. I would expect that at some point there will be an almost perfect C64 core, if that's what you want. Assuming the MISTer interfacing adventure works out, Amiga cores are within sight as well. Oh, I didn't mention that Spectrum and GameBoy cores already run on the machine. Eventually, at some point you must ask yourself – why did I ever want another machine? ... oh, my killing argument? ... the keyboard. Oh My God! The keyboard!!!

If you're broke but curious there actually is a very nice alternative. One that I use myself in my toolchain. The Xemu. Yes, that's among other things, a MEGA65 emulator. It is not perfect by any means and cannot be used for everything. One example of that is that it will only update rasters every line, so if you make an INC \$d021 loop, it will look very different than what you see on the real hardware. If you know that, you will be fine. The emulator has come a long way and it is easy to get up and running. Just download it, get the latest ROM from the community and you're ready to rock. The happy medium in terms of price could be to buy an Atrix board. You'll have some hardware to run things on, but you miss out on the chassis and most of the Commodore connectors though.

Make or break

The team behind the MEGA65 need to make their machine great. This machine was never about making money and the design, the process and the product just REEKS of just that. The one thing we all are driven by - passion!

They should not try to make a machine that works like an old prototype and more importantly one that "has a c64 mode". I watch development discussions and it somehow feels like the team is trying too hard

to be too much. I must remind myself that this is a community driven effort and I need to remind you all of that too. This is not a machine where you should expect everything to work at first boot, but a machine that will grow, just as you grow with it.

Dear team, ignore everything relating to “true c64 mode” or whatever you want to call it. Focus on making the MEGA65 dream machine. We already have a functioning c64. The MEGA65 is incredible as it is, and I have no doubts that any c64 passionate will love any minute with this machine. If you have a c64 in your home today, you’ll want a MEGA65 right next to it. You’ll truly understand it when you, like I did, start exploring it. This will be the most amazing conclusion of all things we are, because I don’t think we will see another milestone as big as this machine in our lifetime. Seeing old game companies like Thalamus and new companies focusing on retro games, I also think the MEGA65 can potentially become the go-to retro gaming platform for software houses and indies who want to make games. Yeah, it’s that good and no. I sincerely hope that a new generation of retro degenerates will surface, but I fear we are the last of our race. Voxels have taken over the world as we know it. The MEGA65 is our swan song and everybody, no matter what part of the Commodore scene you are in, are welcome. It won’t be perfect at birth, but neither was the c64. Remember this is a community driven effort, so you have the power to make changes and improvements. It is entirely up to you to yield that power.

In the beginning of this article, I mentioned that real people are working on this. I’ve spoken with most of the team on different occasions and believe me, this is not about money. This is about passion, and I wish they would make some money off this, because their dedication is monumental. Unfortunately, they won’t, because the way the project is set up and with the price it has, there IS no profit. What you pay for is the production and the components. Production is in Europe and the parts are top shelf components. This is not a cheap production and quality is expensive. It’s like saying that Tesla is an expensive car I can get another car a lot cheaper – you can indeed, but not the same quality.

The MEGA65 is not about making money. It’s about making history. With this article I hope you see that this machine is not only an option, but the natural evolution of the c64. This is the next level.

Who are people that started this project, you might ask? They are sceners, just like you and me. Highlander/Fairlight, Deft/MEGA/TRSi, doubleflash/MEGA and seriously/MEGA. You are awesome and eventually when time matures, the retro scene will bow to you. Bonzai salutes you. We love you for doing what you do, and you have our full support. There are a lot more people involved in the project today. Very passionate and clever people. This being a community driven effort, more people are always welcome. I know for a fact that right now they could use more knowledge in the sound department. So why not join in on the fun and contribute?

The 8-bit scene will live forever!

References

The main MEGA65 page: <https://www.mega65.org/>

Official file hub (get your ROMS here): <https://files.mega65.org>

Xemu: <https://github.com/lgbllgb/xemu>

What is FPGA https://en.wikipedia.org/wiki/Field-programmable_gate_array