Chapter 2
The Commodore 64 and Its Architecture

Abstract Here the unveiling of the C64 is discussed together with its original architecture, introducing the main components that made it unique: the 6510 CPU, the VIC-II and SID chips.

Keywords CES · Vertical integration · Computer architecture · 6510 · VIC-II · SID

Plans for the new machine that would ultimately become the Commodore 64 started hastily already in early 1981 and, by successfully meeting the seemingly impossible deadlines imposed by Jack, Commodore engineers were able to showcase the new computer prototype at the winter Consumer Electronic Show (CES) in Las Vegas in January 1982, alongside all the new material developed for the VIC-20, including the 1600 VICModem, the first inexpensive modem device that was also going to play a pivotal role in popularizing online activities in the years to come, as we will see in Chap. 7.

The showing of the new machine, which could use existing VIC-20 peripherals, was completely unexpected (indeed, not even many people at Commodore itself were aware of its development which was kept under the uttermost secrecy by Jack) and made a terrific impression thanks to several demos emphasizing its high end specs which included 64 KB of overall memory (direct competitors at the time had at most 48 KB), screen resolution able to comfortably display 40 columns of text (the VIC-20 could only display 22, making it unsuitable for serious word processing applications) and advanced multimedia qualities.

In August 1982 the C64 was ready for its commercial debut retailing at $595, a price tag that was thought to be impossible by the competition when originally announced at CES. Indeed, such a low price was only possible thanks to Jack's foresight in implementing a vertical integration manufacturing process centered around the original MOS acquisition done years earlier and it actually left Commodore with a very healthy profit, allowing the company to start pushing down costs even further soon after launch. This started a savage business war where Commodore could continuously slash retail prices to undercut competition

R. Dillon, *Ready*, DOI 10.1007/978-981-287-341-5_2

Fig. 2.1 The original Commodore 64 "breadbin" model in all its glory

while remaining profitable. The C64 (Fig. 2.1) was soon much cheaper than any other competitor, allowing Commodore not only to drastically increase its user base month after month but also to push many other home computer manufacturers into serious trouble or even force them entirely out of business like in the case of Texas Instruments and its TI-99/4A computer.

With worldwide distribution starting in early 1983, the C64 was an instant and ever lasting success that was discontinued only in 1994 once Commodore had to file for bankruptcy.

Overall, according to Commodore's 1993 Annual Report, the C64 sold 17 million units across its different iterations, like the C64C (an early example of what would be seen today as a "slim" hardware revision), SX-64 (a self contained and portable version of the computer), C64G (a later, cheaper breadbin version) and the C64GS (a C64 turned into a game console by removing the keyboard).

The Commodore 64 was a machine that refused to die and be forgotten, imposing itself as an 8 bit icon representative of a whole technological era. As we will see throughout the book, it affected the history and development of games and personal computing in every imaginable way across the years, inspiring people to innovate in several different areas.

But how did it do that? To understand its long lasting influence and how it managed to accomplish all this, we should start by having a quick look at its architecture and see what made it unique.

In its simplest form, like all modern computers, the C64 can be described in terms of the so-called Von Neumann architecture (Fig. 2.2), named after the mathematician John von Neumann (1903–1957).

As we can see from the diagram, a computer can be subdivided into three main parts:

- a **Memory**, to store data and instructions.
- **Input** and **Output** mechanisms to interact with the outside world.

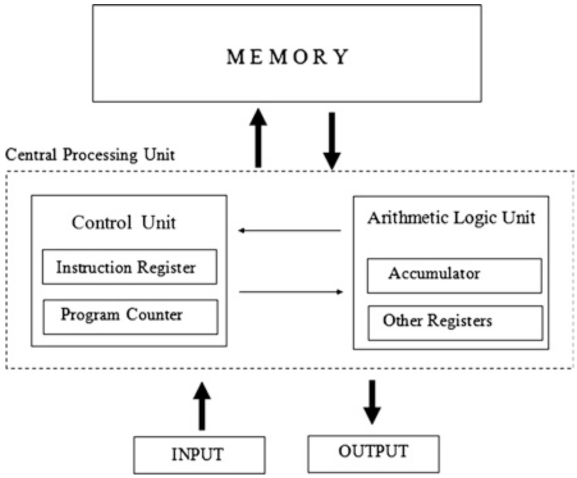

Fig. 2.2 A schematic diagram outlining the components of the Von Neumann architecture

- a **Central Processing Unit** (CPU), whose main components are the **Arithmetic Logic Unit** (ALU), able to perform basic arithmetic and logic operations on data which are retrieved from memory and stored in a few dedicated registers for manipulation (the most important of which is traditionally called "Accumulator") and a **Control Unit** (CU) which implements the instruction set of the CPU and includes two special registers named **Instruction Register** (IR) and **Program Counter** (PC) to keep track of the instruction being executed and the one to execute next.

All of these components are then connected with each other via a bus system that allows data to be transmitted back and forth as needed.

Regardless of the model and origin, all CPU activities can be summarized as a sequence consisting of the following phases:

- **Fetch**: where an instruction stored in the memory location specified by the Program Counter is retrieved and stored in the Instruction Register. When done, the Program Counter is also updated with the memory address of the next instruction to load once the current one has been executed.
- **Decode**: where the Control Unit, holding the fetched instruction in its Instruction Register, dissects it to understand what needs to be done.
- **Execute**: where the CPU actually executes all the steps required by the specific instruction, for example by loading numbers into the various ALU registers to perform different operations, like an addition, whose result is then stored temporarily in the Accumulator for later use by a following operation.
- **Writeback**: where, as the name implies, the final result of the instruction is written back into a memory location or sent to an output device. After this, a new a fetch phase starts to load the next instruction and the whole sequence is repeated till the whole program has been executed.

The CPU mounted in the Commodore 64 was a 6510 (Fig. 2.3), a variation of the original 6502 which allowed for different configurations of the memory layout and

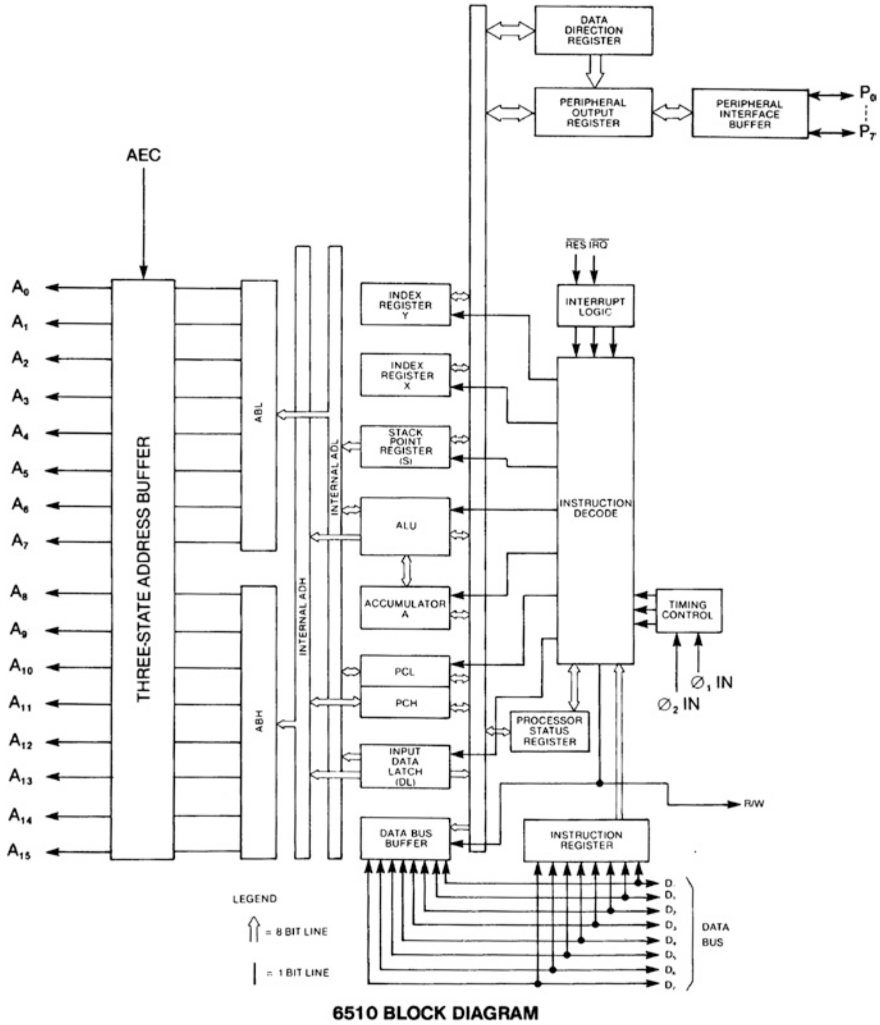

Fig. 2.3 The 6510 CPU block diagram where we can appreciate all its different components

for adopting a "tri-state" address bus, a modification useful for connecting or disconnecting different circuits. This feature was needed, for example, for granting the video chip direct access to the memory, disconnecting the RAM from the CPU itself.

A flexible and enhanced CPU like the 6510 was paramount to the success of the C64 but was not the only reason the new machine raised so much interest at the original unveiling during the 1982 CES, nor of its everlasting success. In fact, two other MOS custom chips were included to set the C64 apart from the rest of the competition: the Video Interface Controller (VIC-II) and the Sound Interface Device (SID).

To really appreciate the uniqueness of the machine, we should gain a little more understanding of what made these components so special.

2.1 The VIC-II

The follow up to the VIC-20's own VIC chip, the VIC-II[1] was designed primarily by Al Charpentier and Charles Winterble by taking into consideration all the strong points of existing graphic chips in competing machines, like native sprite support from the TI-99/4A or collision detection from Mattel's Intellivision, and then synthesize a new, superior one.

In the end, about 75 % of the VIC-II surface was dedicated to implement sprites related features. The chip also allowed for smooth scrolling across a screen resolution of up to 320 × 200 pixels where different colors, from a fixed palette of 16, could be displayed at once. As we see, arcade quality graphics for games were clearly planned from the very beginning of the design stage, with sprites, referred to in the technical documentation as "Movable Object Blocks" (MOB), being a top priority.

Programming wise, working with the VIC-II chip essentially revolved around manipulating its unique 47 registers, which handled all the information related to sprites X and Y coordinates on the screen, their colors, background and border colors, collision detection, the raster counter and more.

It is very interesting to note that, while the chip by default allowed for up to 8 sprites (each being 24 × 21 pixels if monochrome, 12 × 21 pixels otherwise) to be easily handled concurrently per scanline, this limitation could actually be overcome by a smart use of the provided raster interrupt routines, allowing for many more sprites to be displayed on the same screen at once. This technique, usually referred to as "Sprite Multiplexing", worked by triggering an interrupt to pause the drawing process when a given scan line was reached and then call a routine to modify and reload the VIC-II chip's registers as needed before proceeding in drawing the remaining part of the screen: by doing so, already drawn sprites could be relocated and drawn again in a different area of the screen, colors could be added, graphics modes changed and so on, allowing for a much richer and varied experience that could have been impossible otherwise (Fig. 2.4). Theoretically, with each sprite needing 64 bytes of memory to be stored and having the VIC-II 16 KB of overall addressable space for screen, character and sprite memory, up to 256 sprites could be displayed on a C64 screen!

Anyway, adding sprites was not the only trick possible and, as developers got more and more knowledgeable about all the different VIC-II nuances, understanding also how these could be manipulated by using specific quirks of different TV standards like PAL and NTSC, very impressive graphical effects like color

[1] IC Numbers: NTSC: MOS 6567/8562/8564, PAL: MOS 6569/8565/8566.

Fig. 2.4 Katakis, a graphically impressive space shooter released in 1989 by Rainbow Arts, used sprite multiplexing to the maximum effect, displaying many different sprites at once to create a frantic action experience

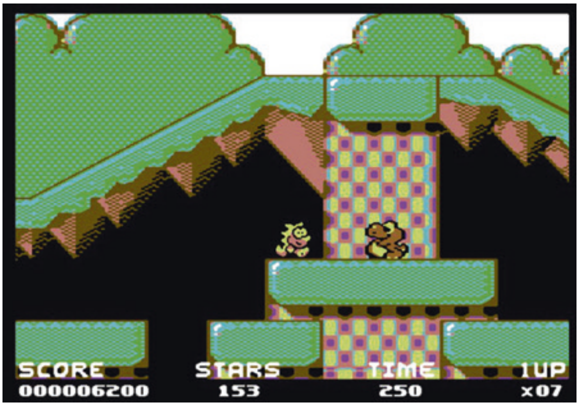

Fig. 2.5 Mayhem in Monsterland. Released in 1993 by Apex Computer Productions, Mayhem was one of the very last titles released on the C64. It was also one of the most technically impressive ever produced thanks to very advanced graphics enhancements, including some that were only possible on PAL systems, like PAL-colorblending, and made the game look almost like a 16-bit production

blending and color interlace could be achieved to show extremely colorful graphics and characters that were hard to match on any other 8 bit computer or gaming system (Fig. 2.5).

2.2 The SID

The SID was, arguably, the most beloved chip in the C64 and it is fondly remembered by many even today.[2] What made it so special?

[2] IC Numbers: MOS 6581/6582/8580.

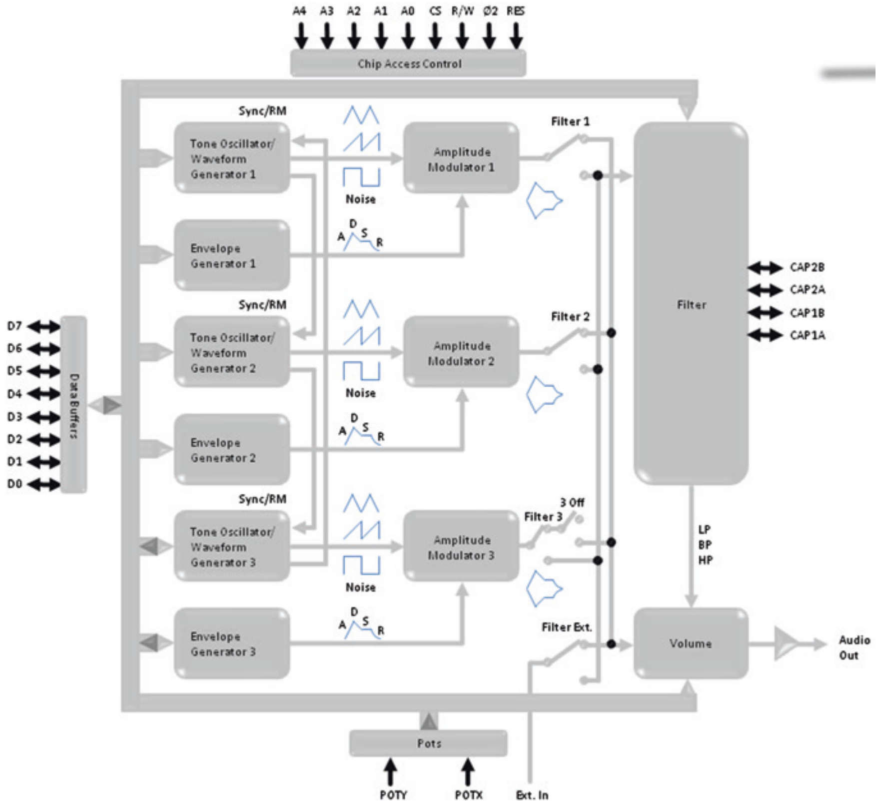

Fig. 2.6 SID block diagram

Designed by Robert Yannes, it was a very advanced sound synthesizer for the time and the first sound chip sporting an envelope generator to be integrated in a home computer.

In particular, the SID made available three synthesizer voices that could be used independently or in conjunction with each other to create complex sounds. Each voice consisted of a Tone Oscillator/Waveform Generator, an Envelope Generator and an Amplitude Modulator (see Fig. 2.6). Manipulating the tone oscillator allowed for fine control of the pitch across a very wide frequency range (8 octaves, approximately from 16 to 4,000 Hz) while the amplitude modulator of the oscillator could dynamically control the volume of the sound according to the input from the envelope generator. The latter could be programmed by specifying different attack/decay/sustain/release (ADSR) values to shape the volume of a note or sound effect in any way the sound designer desired.[3]

[3] More on this in Chap. 3.

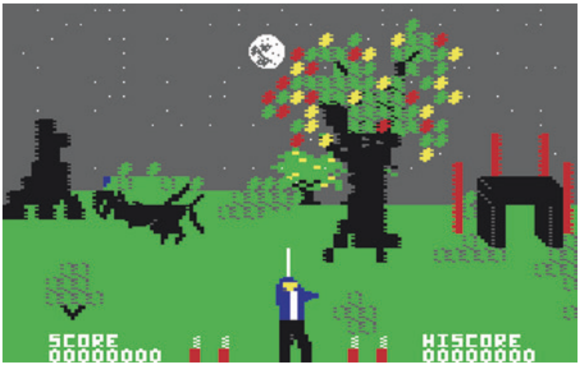

Fig. 2.7 Paul Norman‘s Forbidden Forest [(c)1983 Cosmi] was one of the first titles to push the SID unique capabilities and showcase a dynamic musical score able to follow and haunt the player, creating a truly scary and eerie atmosphere for the game

A programmable filter featuring low-pass, high-pass and band-pass outputs, with 6 dB/octave (band-pass) or 12 dB/octave (low-pass/high-pass) rolloff, was also available for generating even more complex and dynamic tone colors via subtractive sound synthesis.

Last but not least, the SID also had two A/D converters for interfacing with potentiometers and could even process external audio signals, allowing multiple SIDs to be daisy chained or mixed in complex electronic music systems.

All these features together allowed for tremendous flexibility that, in the hands of talented composers and audio programmers like Ben Daglish, Martin Galway, Rob Hubbard, Jeroen Tel and others, pushed gaming audio to new heights: original sound effects and engaging music that could even change dynamically following the gaming action, like in Cosmi‘s 1983 hit “Forbidden Forest” (Fig. 2.7), were finally possible and were a huge step forward compared to anything done before.

Many exceptional sound tracks followed the pioneering efforts of Forbidden Forest, with music constantly gaining more relevance and importance across game development.

Games like Parallax (music by Martin Galway), Delta (music by Rob Hubbard), the Last Ninja (music by Ben Daglish) and Supremacy (music by Jeroen Tel), among others, had music tracks that set them apart from anything else and are still listened to and enjoyed even today, not only when playing the actual games but also via standalone SID playing utilities and emulators.[4]

Now that we have a basic understanding of the C64 main hardware components, we can progress to learn about its actual operation and programming.

[4] Interested readers may check the freely downloadable Java SID Player Music Library: http://sourceforge.net/projects/jsidplay2/.
An online HTML5 emulator is also available at http://www.wothke.ch/experimental/TinyJsSid.html.

http://www.springer.com/978-981-287-340-8

Ready
A Commodore 64 Retrospective
Dillon, R.
2015, XVI, 157 p. 159 illus., Hardcover
ISBN: 978-981-287-340-8