6502/6510 Coding

This section contains articles on programming the 6502/6510 processors in general. The information here is not primarily focused on coding graphics, sound or IO stuff, but covers instead operations of various opcodes and about speed/memory optimization and so on.

Machine-code and Assembly-language

- Beginners guide - Part 1 - Beginners guide to machine-code and assembly-language for the 6510 by Rudi B. Stranden
- Machine language tutorial - An incomplete series about 6510 machine language written by Karmic/HVSC.

Instruction set and Addressing modes

This section should contain info on the instruction set of the 6502 and 6510, addressing modes, cycle tables, illegal opcodes, timings pinout, etc.

General Documentation

- c64doc - detailed specification of opcodes & memory handling, by John West and Marko Mäkelä
- Opcodes and Quasi-opcodes - by Craig Taylor from C=Hacking #1

In general external links should be avoided in the “base” section of this wiki, but the following two links to references of opcodes (including the “illegal” ones) are suitable here:

- Ninjas reference
- Grahams reference

Definitions of constants for Mnemonic to Opcode correspondence

This section contains two lists of constant definitions without meta information for inclusion in your assembler code. The first list includes only the legal opcodes, whereas the second list includes also the illegal ones. The lists were checked against Grahams table, which seems to be error free in contrast to other sources (For example the Commodore Hacking list).

- Legal Mnemonics
- Legal + Illegal Mnemonics

General tutorials

- Tutorials by Craig Bruce, from C= Hacking:
  - Learning ML 1 - Documented and undocumented opcodes.
  - Learning ML 2 - Indexed addressing.

Illegal opcodes

In addition to the standard opcodes available in 6502, the 6510 CPU of the C64 also have a few undocumented opcodes. Some of them are possible to use in a productive way and this section contains some info on this. Note that the fact that these opcodes are not official ones has resulted in a situation where there is no real naming convention. Thus you will find that the same illegal opcode will have a range of different names in different documents.

- Extra Instructions Of The 65XX Series CPU - Overview article by Adam Vardy from 1995 - quite old by now
- No More Secrets 0.92 - Documentation of unintended opcodes by Groepaz (2017-24-12 version).

Illegal opcode tricks

A collection articles and routines that exemplify particular aspects of some illegal opcode.

- Decrease X register by more than 1 - by FTC/HT
- Some words about the ANC opcode - by FTC/HT
- Store X Indexed by Y and Vice-Versa With SHX/SHY - by Cruzer/CML
- Shift bits and throw carry away with ALR - by Cruzer/CML

The Addresses 00 and 01

These two addresses are used for (among other things) settings up the memory layout of the C64. These are the only differences between a 6502 and a 6510 (besides the Tri-State Bus, which is needed to cooperate with the Video Chip)

- Memory Management - 00 and 01 can be used to switching on/off BASIC, KERNAL, and CHAR ROM.
- Datasette
- RAM beneath $00 and $01

General stuff

- CPU Clocking - precise MHz figures
- Detect CPU Type - by UZ (from CC65)

- Common Pitfalls - Several Tips&Tricks of the most used Coding Mistakes
- Using a byte as bitcounter and value container at the same time - useful when dealing with bit streams
- Speeding up and optimising demo routines - by conrad
- Speedcode by Cruzer/CML
- Loops vs unrolled by Bitbreaker/Oxyron^Nuance
- Advanced optimizing - Tricks to save cycles, including use of illegal opcodes - by Bitbreaker/Oxyron^Nuance
- Launching long tasks from IRQ handler - by Bitbreaker/Oxyron^Nuance
- Variable speedcode runlength - by Bitbreaker/Oxyron*
- Practical Memory Move Routines - by Bruce Clark - taken from www.6502.org
- Clearing a Section of Memory - from 6502 Software Gourmet Guide & Cookbook - taken from www.6502.org
- Set a byte to non-zero - by White Flame
- Making a Counter - Example code of an increasing decimal counter - by Scout/Silicon Ltd.
- Threads on the 6502 - an example on how threads can be used efficiently on the 6502 processor.
- Swapping ZP data - small snippet to save/restore ZP data by enthusi
- Dispatch on a byte - Useful in interpreters and decompression, by White Flame
- Rotate byte and act on 1-bits - Invented by Hoogo, written by FTC
- Decoding bitstreams for fun and profit, by lft.
- Protecting against soft-resets - by Karmic/HF