6502/6510 Maths

This is where you find pure math routines, i.e. not algorithms dependent of the workings of the graphics chip such as line drawing (which you find in the VIC section instead). Since the material in here does not really relate to other hardware than the 6502/6510, it is also useful for other systems which are based on 6502/6510 CPUs.

Math Operations

The only mathematical operations included in the 6502/6510 set of instructions are addition and subtraction. In order to do things like multiplication, division, roots, squares and so on, you have to write code for it.

Comparison

- Beyond 8-bit Unsigned Comparisons - A very interesting article by Bruce Clark at www.6502.org
- 8-Bit Ranged Comparison - By White Flame

Addition/Subtraction

- 16bit addition and subtraction - 16-bit addition and subtraction by FMan
- Inverse subtraction - .A = operand - .A, opposite of what SBC does by White Flame
- Signed 8bit + 16bit addition - Sign extension by White Flame

Multiplication

- 8bit multiplication (8bit product) - by White Flame
- 8bit multiplication (16bit product) - also can be 16*8=16bit - by White Flame
- Short 8bit multiplication (16bit product) - by Graham
- 8bit multiplication (16bit product) fast no tables - by djmips
- 16bit multiplication (32-bit product)
- Fast 8bit multiplication(16bit product) (signed multiplication) - by Oswald/Resource
- Seriously fast multiplication (8/16 bit signed/unsigned multiplication) - by JackAsser/Instinct
- Fastest multiplication (16 bit unsigned multiplication) - by Repose
- Another fast 8bit multiplication (16bit product) - by litwr
- Table generator routine for fast 8 bit mul table - by Graham
- Multiplication with a constant - by Various
- 24 bit multiplication (signed or unsigned) -by Neils

Division

- 8bit divide (8bit product)
- 8bit divide by constant (8bit result)

- 16bit division (16-bit result)
- 24bit division (24-bit result)
- signed 8bit divide by 2 (arithmetic shift right) - by Bitbreaker/Oxyron/Nuance

Exponentiation

- Exponentiation

Square Root

- Fast sqrt (extracted from CSDB forums) - Graham
- 16bit and 24bit sqrt (extracted from CSDB forums)

Log

- 8bit logarithm table generator routine - by doynax

Trigonometrics

- 8bit atan2 (8-bit angle) - by doynax
- Generating Sines with BASIC - by Doynax
- Generating Approximate Sines in Assembly - by White Flame

Floating point

- Floating Point Routines for the 6502 - by Roy Rankin (Stanford) and Steve Wozniak (Apple) -
- Errata for Rankin's 6502 Floating Point Routines - by Roy Rankin (Stanford) - from 1976
- Kernal floating point mathematics - Using the routines in the C64 ROM
- Mandelbrot - Example Mandelbrot generator in ASM using the routines in the C64 BASIC ROM

Mathematics in assembly

- Mathematics in Assembly Article Series - A series of 8 articles by Krill/Plush. Brought here on popular demand!

- Numerical systems (binary and hexadecimal explained)
- Basic math operations (addition, subtraction, carry bit, bit shifting, 16 or more bit operations)
- Fixed point arithmethic (introduces the representation of fractional numbers i.e. numbers smaller than 1)
- Two's complement system (introducing signed numbers)
- Multiplication and Division
- Trigonometric functions

The art of 3d

- Rotating and transforming vertices
- Filling the vectors
- Perspective
- Backface Culling

You may also want to read Stephen Judd's awesome series titled “A different perspective” about 3d coding in C= Hacking:

- C= Hacking Issue 8 - August 1994
- C= Hacking Issue 9 - January 1995
- C= Hacking Issue 10 - June 1995

Algorithms

Sorting

The sorting routines below are taken from www.6502.org.

- Bubble Sort (8-bit Elements) - from 6502 Software Design
- Bubble Sort (16-bit Elements) - from 6502 Software Design
- Combination Sort (8-bit elements) - by Daryl Rictor
- Optimal Sort (8-bit elements) - by Mats Rosengren
- Optimal Sort (16-bit elements) - by Mats Rosengren
- Shell Sort (16-bit elements) - by Fredrik Ramsberg
- Quicksort (16-bit elements) - by Vladimir Lidovski aka litwr

Also see the page on sprites for some sorting routines optimised for sprite multiplexers.

Random Numbers

- BASIC RND routine - Stephen Judd
- Fast 8bit ranged random numbers - by kerm1t
- 16bit pseudo random generator
- another 16bit pseudo random generator
- Two very fast 16bit pseudo random generators as LFSR - with nearly 2^31 period - by Hanno Behrens
- 32bit Galois LFSR
- Flexible Galois LFSR - by gregg
- Small, fast 8-bit PRNG - with full 8-bit period - by White Flame
- Small, fast 16-bit PRNG - with full 16-bit period - by White Flame
- 16bit xorshift random generator

Number conversion

- Hexadecimal to Decimal Conversion - using lookup tables, by Garth Wilson - taken from www.6502.org
- More Hexadecimal to Decimal Conversion - without lookup tables, by Andrew Jacobs - taken from www.6502.org
- Another Hexadecimal to Decimal Conversion - using lookup tables, to plain ASCII, by Mace
- Int16 and UInt16 conversion to String
- Decimal to Hexadecimal Conversion - by Mace
- 8 bit to Hexadecimal Conversion - by Abujok
- 32 bit hexadecimal to decimal conversion - no tables, no BCD mode, by Graham
- Tiny .A to 3-Digit Decimal Conversion
- Sign Extension - by White Flame

Packing/Crunching algos

- Streaming 1/2/4/8-bit Numbers Without Spanning Bytes - by White Flame
- The Secret of Fast LZW Crunching - by Antitrack/Legend, Sept.22nd, a.d.1998, for Domination paper edition
- RLE Toolkit for CC65 v 1.0 - RLE pack/unpack coded by MagerValp
- 2Mhz Time Crunch V5 disassembled - Disassembly of Stoat and Tim's Time Cruncher V5. Disassembled by Marko Makela.
- Decruncher for MDG-Packer/Linker 1.1 - Dasm source code for depacking MDG-Packer Files.
- MDG Bytesmasher 0.04 - Dasm Source code of a simple rle packer with 5 packcodes.
- LZMPi compressor and decompressor. Includes source and examples.
- Compression Benchmarks. A comparison of compression ratios and decompression speeds of various packers.

EXOMIZER

- Exomizer level compress/decompression for beginners - by Richard / TND