

Manual for Tali Forth 2 for the 65c02

Scot W. Stevenson

Version ALPHA, 26. June 2018

Table of Contents

Dedication	2
Introduction	3
Why?	4
The Big Picture	4
Overview of Tali Forth	6
User Guide	7
Installing	8
Running	9
The Editor	10
Developer Guide	11
How Tali Forth works	12
Developing	13
Future and Long-Term plans	14
Appendix	15
FAQ	16
Testing Tali Forth 2	17
Thanks	18
References and Further Reading	19
Colophon	20

Tali Forth 2 is a bare-metal ANSI(ish) Forth for the 65c02 8-bit MPU. It aims to be, roughly in order of importance, easy to try out (just run the included binary), simple (STC Forth model), specific (for the 65c02 only), and standardized (ANSI Forth).

Dedication

For the crew at 6502.org, who made this possible in more ways than one.

Introduction

Why?

Forth is well suited to resource-constrained situations. It doesn't need lots of memory and doesn't have much overhead. It can take full advantage of whatever hardware or interfaces exist. [\[CHM1\]](#)

— Charles H. Moore, redgate Hub 2009

The Big Picture

This section provides background information on Forth, the 6502 processor, and why anybody would want to combine the two. It can be safely skipped if you already know all those things.

The 6502 CPU

It is a well-established fact that humanity reached the apex of processor design with the 6502 in 1976. Created by a team including Chuck Peddle and Bill Mensch, it was the engine that powered the 8-bit home computer revolution of the 1980s. [1: Rumor has it that there was another MPU called "Z80", but it ended up being a mere footnote.] The VIC-20, Commodore PET, Apple II, and Atari 800 all used the 6502, among others.



Figure 1. The 65c02 MPU. Photo: Anthony King, released in the public domain

More than 40 years later, the processor is still in production by the [Western Design Center](#). Apart from commercial uses, there is an active hobbyist scene centered on the website [6502.org](#). Quite a number of people have built their own 8-bit computers based on this chip and the instructions there, including a [primer](#) by Garth Wilson. It is for these systems that Tali Forth 2 was created.

The most important variant of the 65c02 produced today is the [65c02](#), a CMOS chip with some additional instructions. It is for this chip that Tali Forth 2 was written.

But why program in 8-bit assembler at all? [2: Wilson answers this question in greater detail as part of his [6502 primer](#).] The 65c02 is fun to work with because of its clean instruction set architecture

(ISA) This is not the place to explain the joys of assembler. The official handbook for the 65c02 is [_Programming the 65816, including the 6502, 65C02 and 65802_ \[EnL\]](#).

Overview of Tali Forth

User Guide

Installing

Running

The Editor



Currently, there is no editor installed. The plan is to include a simple line-based editor in the style of `ed`.

Developer Guide

How Tali Forth works

Developing

Future and Long-Term plans



This section is missing.

Appendix

FAQ

Testing Tali Forth 2

Thanks

References and Further Reading

[CHM1] "Charles H. Moore: Geek of the Week", redgate Hub 2009 <https://www.red-gate.com/simple-talk/opinion/geek-of-the-week/chuck-moore-geek> [EnL] *Programming the 65816, including the 6502, 65C02 and 65802*, FEHLT

Colophon

The Tali Forth 2 Manual was written with the vi editor in AsciiDoc format, formatted to HTML with AsciiDoctor, and version controled with Git.