

# The "Emu 2.0" 8bit RISC microprocessor

M!lk Microcomputer Corporation

1 January 1978



## 1 Information

This Manual describes the assembly language format, and how to write assembly language programs for the "Emu 2.0" microprocessor. Detailed information on the operation of specific assemblers is available in the Operator's Manual and Installation Guide for each specific assembler.

## 2 Processor Capabilities

### 2.1 Overall Description

The "Emu 2.0" is an 8bit RISC microprocessor. It has two registers, the first being A (Accumulator), an 8bit register used for arithmetic and logic calculations. The second register is called PC (Program Counter), which is a 12bit register used to point the address in RAM of the current instruction that the microprocessor needs to execute.

## 2.2 Memory

The microprocessor makes use of a full 12bit Address Space for all its operations, including storing the instructions it needs to execute and the data it needs to process. Each memory address is defined as being between 0x000 and 0xFFF.

## 2.3 Instruction Set Architecture (ISA)

Each instruction has a fixed 2-byte (16bit) length. The instructions, represented as hexadecimal (where X can be any value between 0 and F) are listed below, divided by their role:

### 2.3.1 Arithmetic

Hex Code	Description
00 XX	Add XX to A and store the result in A
01 XX	Set A = XX
02 XX	Xor A with XX and store the result in A
03 XX	Or A with XX and store the result in A
04 XX	And A with XX and store the result in A
8X XX	Set A = [XXX]
DX XX	Xor [XXX] with A and store the result in [XXX]
FX XX	Set [XXX] = A

Note: [XXX] Represents the Byte at address XXX

### 2.3.2 I/O

Hex Code	Description
13 37	Send A to Serial Out

### 2.3.3 Control Flow

Hex Code	Description
2X XX	Jump to Address XXX
3X XX	Jump to Address XXX if A = 0
4X XX	Jump to Address XXX if A = 1
5X XX	Jump to Address XXX if A = 255
60 XX	Compare A to XX and store comparison result in A
7X XX	Compare A to [XXX] and store comparison result in A
BE EF	Jump to 0x100 and set A = 0x42

Note: Comparison results are 0 if A = XX; 1 if A < XX; 255 if A > XX

### 2.3.4 Security

Hex Code	Description
9X XX	Block Writes to [XXX]
AX XX	Unblock Writes to [XXX]
CX XX	Frobnicate [XXX] and store the result in [XXX]

Note: The Frobnicate operation Exclusive-ORs the given byte with 0x42.

### 2.3.5 Misc

Hex Code	Description
EE EE	No Operation

Note: A known quirk of this microprocessor is that any undefined instruction has the unique effect of decrementing register A.

## 2.4 Startup And ROM Execution

At boot, before executing the ROM, the microprocessor sets A to 0 and PC to 0x100. The bits from the entire address space are then set to 0, and the ROM contents are being loaded starting from address 0x100. All bytes in memory start as being Write-Unblocked. The ROM is now being executed.