

基于 BrainFuck 架构的开发板

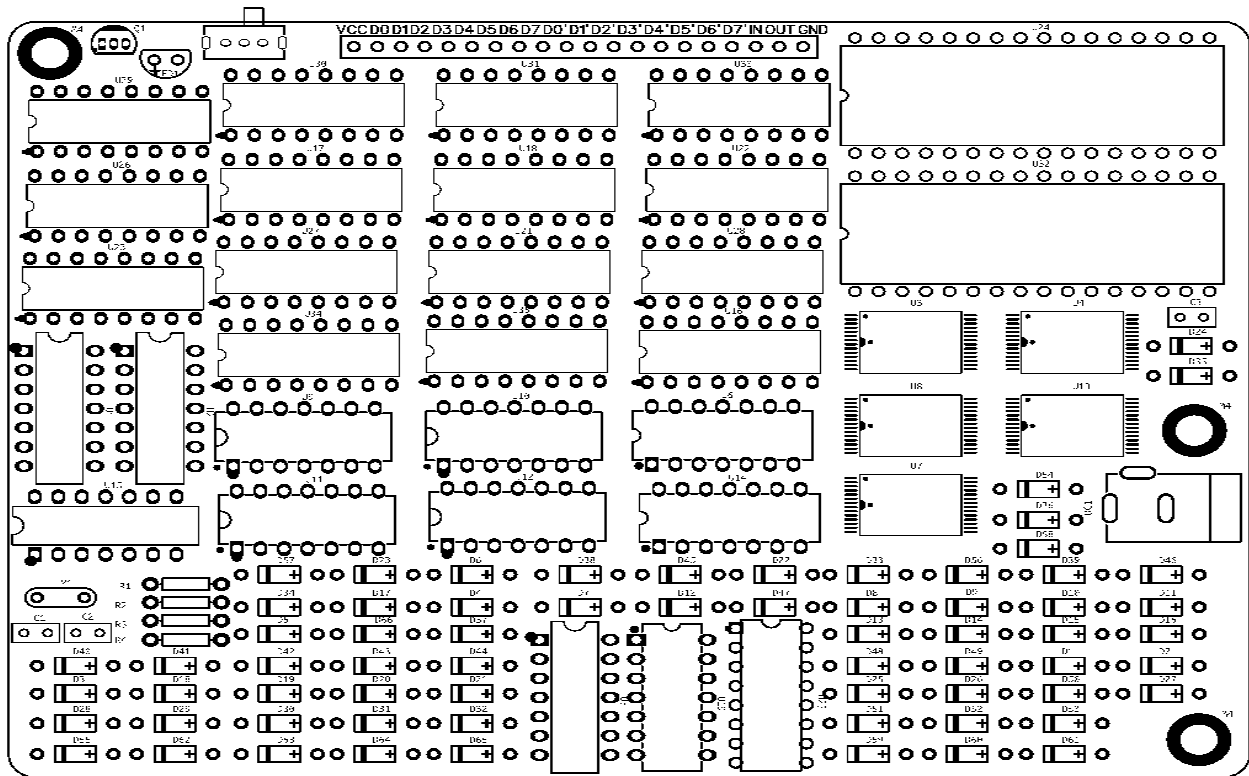
概述

BFT Machine 是一块高速、纯 TTL 逻辑电路而廉价的开发板，处理单元基于 BrainFuck 架构，属于通用计算机。

特性

- 拥有 1MiB 大容量高速可擦写只读存储器，以便烧写程序。
- 拥有 4KiB 可读写内存，另附 4KiB 栈存储器。
- 板面积较小，带有三个定位孔，可作固定处理。
- 速度较快，可支持 16Mhz ~ 25Mhz（极限）晶振。
- 元件好找，替代件丰富，经济费用较低。
- 附带扩展槽，可进行后续输入/输出板开发。
- DC +5V 供电。

外观图



输入/输出引脚说明¹

引脚编号（从左到右）	功能
1	DC +5V 电源
2~9	数据输入引脚（SD0~SD7）
10~17	数据输出引脚（D0~D7）
18	由输入指令触发的信号（IN）
19	由输出指令触发的信号（OUT）
20	GND 接地

¹ 除非引脚后带有 # 符号，所有引脚都为高电平有效。

BrainFuck 机器模型

BrainFuck 是由 Urban Müller 于 1993 年创建的。其设计了一种最小化机器模型，其应当具有：

- 支持顺序读写的存储结构作为内存
- 输入/输出流

经我改进的机器模型仅具有 13 种操作，如：

- 转移到下一个内存对象（用 > 符号表示）
- 转移到上一个内存对象（用 < 符号表示）
- 将当前的内存对象自增（用 + 符号表示）
- 将当前的内存对象自减（用 - 符号表示）
- 输入当前内存对象到输入流（用 , 符号表示）
- 输出当前内存对象到输出流（用 . 符号表示）
- 将栈顶的最高 4 位记录进去（用 A 符号表示）
- 将栈顶的 15~8 位记录进去（用 B 符号表示）
- 将栈顶的 7~0 位记录进去（用 C 符号表示）
- 跳转到栈顶地址 + 1 的位置（用 D 符号表示）
- 如果为零，跳转到栈顶地址 +1 的位置（用 E 符号表示）
- 添加一个新的栈顶（用 F 符号表示）
- 弹出栈顶（用 G 符号表示）

可以证明这样的机器模型是图灵完备²的。

指令集

一条有效的指令应当是 8 位的。

第 4 ~ 6 位为操作编号，对应 8 种操作。

第 0 ~ 3 位为重复次数，表示将要重复多少次（1 ~ 17 次）

指令								描述
1	0	0	0	Count	Count	Count	Count	+
1	0	0	1	Count	Count	Count	Count	-
1	0	1	0	Count	Count	Count	Count	<
1	0	1	1	Count	Count	Count	Count	>
1	1	0	0	Count	Count	Count	Count	,
1	1	0	1	Count	Count	Count	Count	.
1	1	1	0	Data	Data	Data	Data	A
1	1	1	1	Count	Count	Count	Count	B
0	0	0	1	Count	Count	Count	Count	C
0	0	1	0	Count	Count	Count	Count	D
0	0	1	1	Count	Count	Count	Count	E
0	1	0	0	Count	Count	Count	Count	F

² 在可计算性理论中，如果一系列操作数据的规则（如指令集、编程语言、细胞自动机）可以用来模拟任何图灵机，那么它是图灵完备的。

0	1	0	1	Count	Count	Count	Count	G
0	1	1	X	Count	Count	Count	Count	NOP
0	0	0	0	Count	Count	Count	Count	NOP

其中，正如上文所言，Count 代表指令重复次数，各寄存器都是**循环溢出**³的。

输入/输出与扩展插槽应用

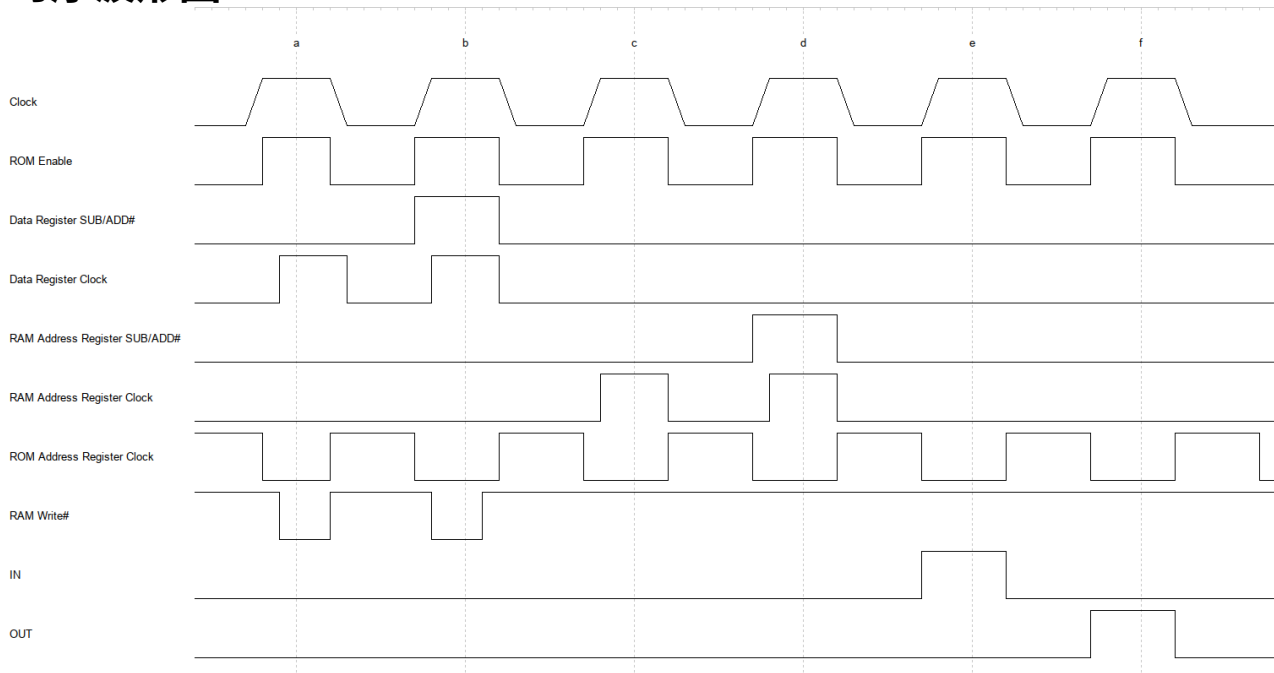
当执行，操作时，数据寄存器允许来自 SD0 ~ SD7⁴的输入，并在 IN 上输出一个脉冲，这个脉冲直到下次指令才会消失，因此，可以通过连续的，操作在该引脚上输出连续脉冲。

当执行. 操作时，OUT 上会输出一个脉冲。当然，即使没有这个脉冲，外部设备也被允许直接通过 D0 ~ D7 读取当前内存内容。

因此，处理器的输入/输出完全转交给了扩展插槽。

下一步的计划是设计扩展卡，以支持串口、键盘、视频、声音等输入/输出。

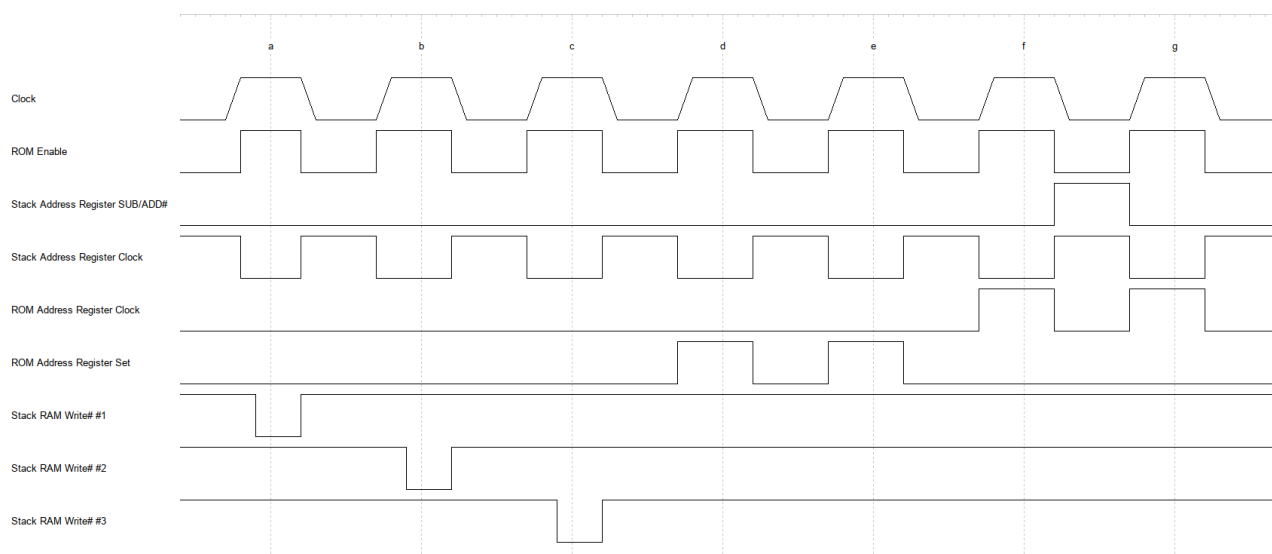
时序波形图



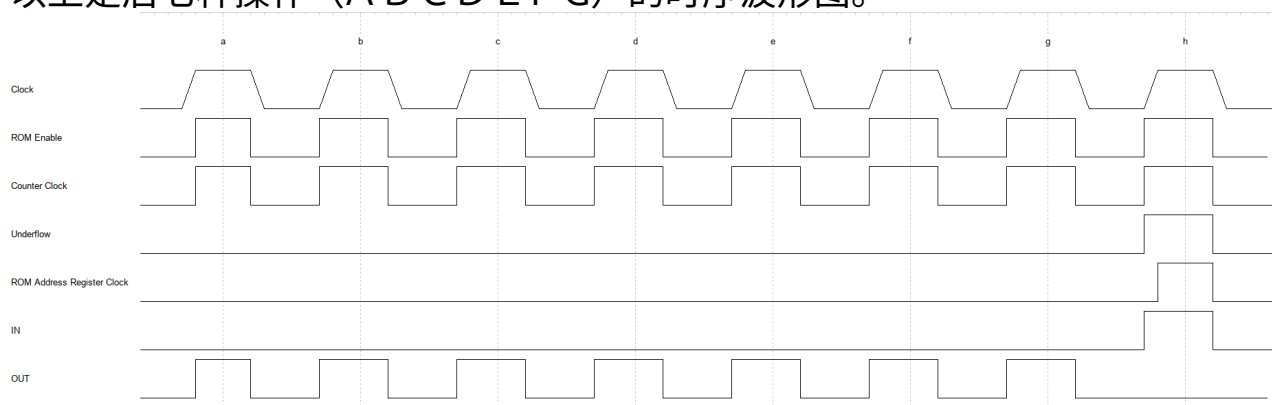
以上是前六种操作（+ - < > , .）的时序波形图。

³ 此处指有限存储空间下，超出空间表示范围而回到起点的现象，例如 8 位寄存器下，255+1=0 的情况。以此道理，如果程序执行完毕没有死循环加以阻止，就会进入错误的流程。

⁴ 参见上文输入/输出引脚说明。



以上是后七种操作（A B C D E F G）的时序波形图。



以上是执行 7 次 . 操作，紧接 1 次，操作的时序波形图。

它们的指令如下：

1	1	0	1	0	1	1	0
1	1	0	0	0	0	0	0

第一条语句，1101 表示输出操作，0110 表示执行 $(6 + 1) = 7$ 次操作。

第二条语句，1100 表示输入操作，0000 表示执行 $(0 + 1) = 1$ 次操作。

Counter Clock 是程序计数器时钟，当执行下一条语句时开始初始化，累减至下溢为止。其作用为控制程序地址增加，当时钟执行而地址未增加时，就会重复执行程序。