第一次在C语言上使用了滤波器，之前都是MATLAB模拟，从没想过在软件上实践这个。这其中最核心的东西就是传递函数不仅仅是FPGA硬件上设计用，软件上代码也可以用。这才是真正意义的硬件和软件的切换吧。传递函数可不仅仅是对应着Z域或者S域的频域，还对应着时域。延时器就是在保存上一个时刻的值，乘法器就是在软件上做一个相乘。加法器做相加。那你可能会问，为什么我们还要做硬件的乘法器和加法器。因为软件上的这些处理受限于单片机时钟频率，速度不会很高。不像硬件，可以做到很高的频率。而FPGA就是一个神奇的东西，仍然可以自主编程，但是最后做这些工作的不是软件，而是硬件。恩好吧单片机也是这样，只不过FPGA给了你更高的自由罢了。FPGA真正的优势在于它的高频率，频率轻轻松松上百兆，上G也不是梦想，那么为什么不拿Intel的芯片做嵌入式呢？那个频率就是以G为单位了？功耗，开源？

FIR的C实现相对比较简单，不对，写系统的单位冲击响应简单，求出滤波器的冲击响应之后呢？y=x\*h。输出就是输入和系统冲击响应的乘积吗？时域上自变量是时间，就是某一时刻的时域强度乘以系统任意时刻的强度相加吗？或者是输入所有时刻的和系统某一时刻的强度相乘相加当做这一时刻的输出。频域上就是直接相乘就是结果了。可是差分方程的意义上是什么呢？不应该差分方程才是在软件上能够实现的吗？传递函数的结构可以直接在硬件上实现，这就是FPGA，或者DSP处理器。那么在软件上呢？也可以相乘相加呀。而且，即使硬件上是模块，可是到时候走的也还是数字信号不是吗？

可以继续手写IIR，FIR的滤波器C，结合考虑FPGA或是DSP的硬件，弄清楚 传递函数，单位冲击响应，差分方程的联系和应用，以及转换。

传递函数就是差分方程，微分方程！！！而单位冲击响应可以求传递函数，就可以求微分方程，差分方程。

不仅仅是那几个模拟滤波器，其他模拟滤波器都可以从S域变换到Z域，在数字上进行解决。比如RC滤波器，RC也是IIR的一种，但是不是巴特沃兹，所以是我一开始的出发点错了。然后就是RC的差分方程，就是那样的，但是系数怎么确定？我之前的是定性，而它是定量。但是其实最终的结果是类似的，因为二者都有一个问题就是灵敏度和响应速度的问题。但是还有几个问题，一个就是那个巴特沃兹为什么没有用？MATLAB试一下呢。还有那几个新的映射的办法？这和互补滤波有什么关系？其实计通网那个不过也就是一个滤波吧，对数据。