DDT 表计算：

输入为 8bit，即有 256 种可能。对每一种输入x找出与其差分为$\triangle x$的x_i，分别计算其输出y、y_i以及$\triangle y$。然后将对应的$(\triangle x,\triangle y)$的值+1，将所有可能的输入计算完以后即得到 DDT 表。

LAT 表计算：

输入为 8bit，输出也为 8bit，那么对应的线性关系一共有$2^8 * 2^8 = 256 * 256$种。输入 x 一共有 256 种，对应的输出 y 也有 256 种，对于每一种输入输出都有 256*256 种线性关系，计算出每个输入输出对应的某个线性关系满足，若相等，则在对应的表项中+1。例如输入 x 为 00000001，输入 y 为 01110010，满足线性关系 $x_1 \wedge x_2 = y_1 \wedge y_2$，（下标为对应的二进制位 0 或 1）则将对应的(00000011,00000011)的值+1。（只要某个下标 i 出现，则对应的坐标第 i 位就为 1，否则为 0）

实验结果见 zuc_sbox.txt，源代码见 ZUC_S0&&S1.py

为什么最后一轮的 key mixing 不能去掉？

最后一轮中的 key mixing 不能去掉是因为 S 盒是可逆的，如果没有 key mixing 做保护，就可以直接通过 S 盒的输出逆向得到 S 盒的输入，这样最后一个 S 盒也失去了意义，相当于整个加密体系少了一轮加密，密码的强度降低了。