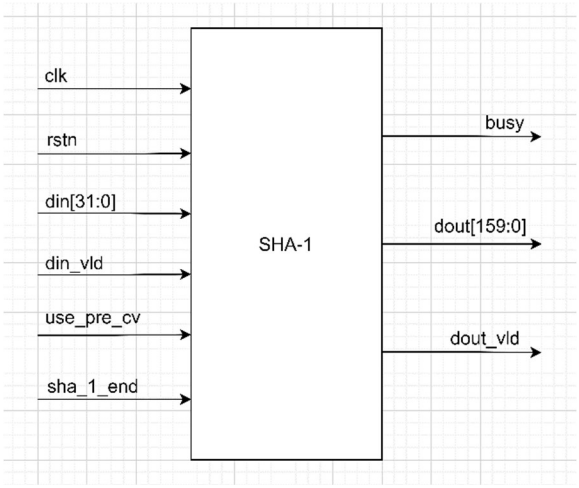
SHA-1 哈希加密算法

SHA-1 是一种哈希算法，对给定的输入数据，按照 SHA-1 规定的算法，计算 160bit 哈希值。输入数据按 512bit（16*32bit）拆分为 message block，按照 message block 计算哈希值。

输入有效 message 的末尾 0x80，最后一个 block 的最后 64 位表示有效的 bit 长度（不包括 0x80）。



clk	时钟信号；
rstn	复位信号；
din[31:0]	待加密数据输入信号，有效 16 个周期；
din_vld	输入数据有效使能信号，有效 16 个周期；
use_pre_cv	当前计算是否使用上一次的计算结果；
sha_1_end	当前计算是否为最后一个 message block；
busy	当前模块计算中；
dout[159:0]	160bit 加密数据输出；
dou_vld	加密数据输出使能信号；

80 轮(t)计算：

A_next = S^5(A) + f(t;B,C,D) + E + w(t) + k(t);
A = A_next; B = A; C = S^30(B); D = C; E = D;

输出： H0 = H0 + A,
H1 = H1 + B,
H2 = H2 + C,
H3 = H3 + D,
H4 = H4 + E;

初始时：　H0 = 67452301，
H1 = EFCDAB89，
H2 = 98BADCFE,
H3 = 10325476,
H4 = C3D2E1F0;

f(t)每轮计算变换函数：

f(t; B, C, D) = (B & D) | ((~B) & D);	0<=t<=19，
f(t; B, C, D) = B ^ C ^ D;	20<=t<=39，
f(t; B, C, D) = (B & D) | (B & D) | (C & D);	40<=t<=59，
f(t; B, C, D) = B ^ C ^ D;	60<=t<=79;

k(t)每轮变换：

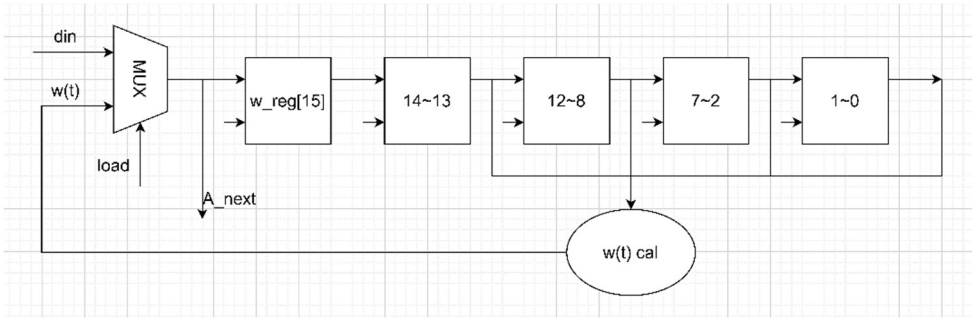
k(t) = 5A827999,	0<=t<=19，
k(t) = 6ED9EBA1,	20<=t<=39，
k(t) = 8F1BBCDC,	40<=t<=59，
k(t) = CA62C1D6;	60<=t<=79;

S^m：循环左移 m 位。

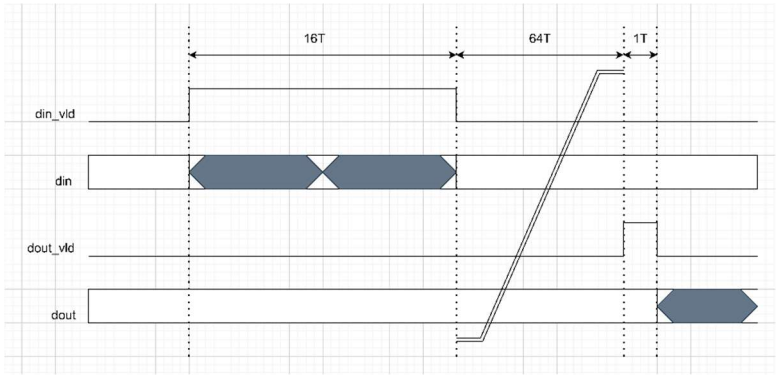
w(t)：输入更新

w(t) = din;	0<=t<=15，
w(t) = S^1 (w(t-3) ^ w(t-8) ^ w(t-14) ^ w(t-16));	16<=t<=79，

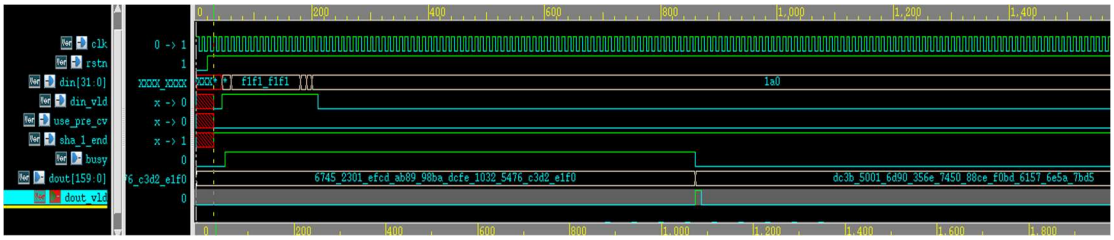
w(t) HW 实现——移位寄存器：



总体输入输出波形：

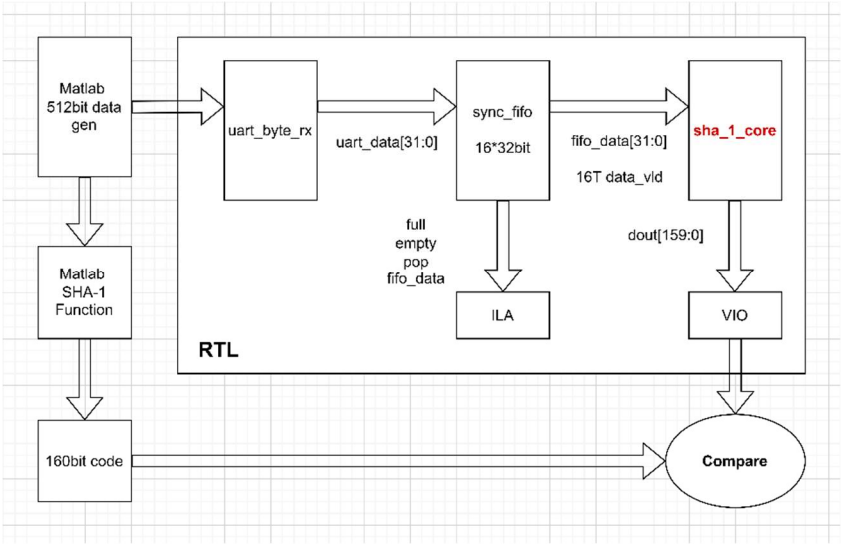


RTL 验证波形：

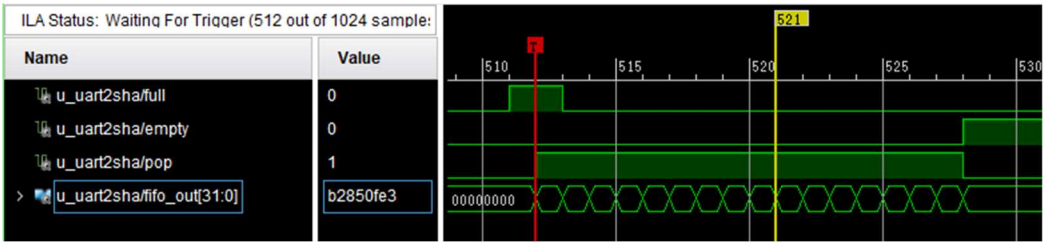


Matlab&FPGA 联合验证：

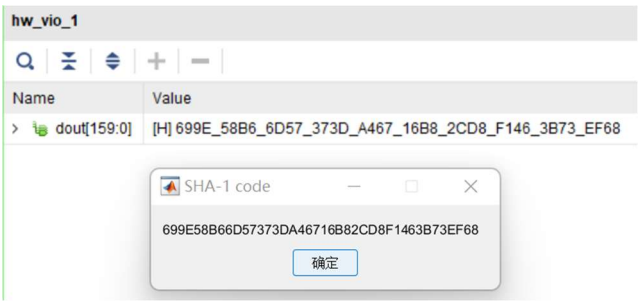
Matlab 联合 FPGA 仿真的流程示意如下图所示：



ILA 验证波形：



VIO 输出与 Matlab 模型运行结果对比（随机一组 512bit 数据）：



GitHub: https://github.com/Jon3Y/SHA_1.git