SOPC-based Design and Realization of a Video Chaotic Secure Communication System

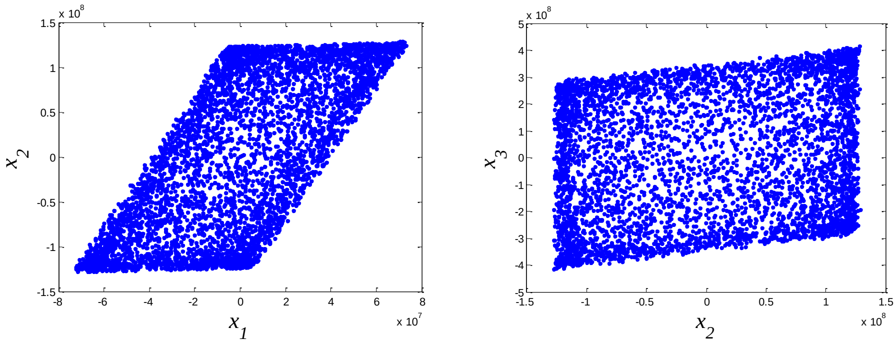
In the video chaotic secure communication systems, there exist two kinds of hardware realization modes, namely embedded and non-embedded ones. The embedded mode is mainly implemented on ARM or DSP-based platform, while the non-embedded mode is always implemented on FPGA-based platform. Hardware experiments have demonstrated that the embedded mode can't well satisfy the real-time requirement due to the serial operating manner. In contrast, the non-embedded mode can well satisfy this requirement through the parallel operating manner of FPGA, but it has some disadvantages for hardware realization, especially difficult for TCP/IP protocol, Ethernet sending and receiving, which is realized by Verilog HDL. To cope with the above-mentioned challenge, this paper proposes a new approach for hardware realization based on system on a programmable chip (SOPC) technology. The main feature of the SOPC-based scheme is that it can make full use of both FPGA and ARM, where the FPGA is used for video capturing, displaying, encrypting and decrypting, and the ARM are adopted for TCP/IP protocol, Ethernet sending and receiving. SOPC technology gives full play to the advantages of both FPGA and ARM complement each other, which leads to higher real-time performance and convenient to achieve hardware realization. In order to improve the ability of resisting divide-and-conquer attack and chosen-ciphertext attack, a 3-dimensional chaotic cipher algorithm utilizing the low 8 bits derived from multiplying is designed for RGB tricolor encryption-decryption. Hardware realization results verify the feasibility and effectiveness of the proposed SOPC-based approach.

In order to take both the required hardware resources of FPGA and a certain degree of security performance of the chaotic cryptographic algorithm into account, the size of system's dimension n should be appropriately selected. In the case of selecting higher-dimensional chaotic systems, although the system's safety performance can be enhanced, but more FPGA hardware resources are also required, and vice versa. In our SOPC-based hardware experiment, considering the limited hardware resources allocation on FPGA, it is appropriate for selecting a 3D chaotic system to balance both the required hardware resources and a certain degree of security performance. Based on the above-mentioned considerations design a 3D chaotic system, such that

$$\begin{cases} x_1(k+1) = a_{11}x_1(k) + a_{12}x_2(k) + a_{13}x_3(k) \\ x_2(k+1) = a_{21}x_1(k) + a_{22}x_2(k) + a_{23}x_3(k) \\ x_3(k+1) = a_{31}x_1(k) + a_{32}x_2(k) + a_{33}x_3(k) + \varepsilon\sin(\sigma x_1(k)) \end{cases}$$

where $g(\sigma x_1(k),\varepsilon) = \varepsilon\sin(\sigma x_1(k))$ is a uniformly bounded feedback controller, $\varepsilon = 3.3\times10^8$ and $\sigma = 2.5\times10^5$ are the control parameters, and the nominal matrix A is given by

$$\boldsymbol{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} = \begin{pmatrix} 0.09 & -0.37 & 0.1 \\ -0.1 & -0.18 & 0.37 \\ 0.27 & -0.27 & 0.19 \end{pmatrix}$$



(a) $x_1 - x_2$ plane (b) $x_2 - x_3$ plane

Fig. 1. Phase portrait of the chaotic attractor

The corresponding chaotic encryption algorithm for chaotic encryption module designed as:

$$\begin{cases} x_1^{(e)}(k+1) = a_{11}x_1^{(e)}(k) + a_{12}x_2^{(e)}(k) + a_{13}x_3^{(e)}(k) \\ x_2^{(e)}(k+1) = a_{21}p(k) + a_{22}x_2^{(e)}(k) + a_{23}x_3^{(e)}(k) \\ x_3^{(e)}(k+1) = a_{31}p(k) + a_{32}x_2^{(e)}(k) + a_{33}x_3^{(e)}(k) + \varepsilon \times \sin(\sigma \times p(k)) \end{cases}$$

The corresponding chaotic encryption algorithm for chaotic encryption module designed as:

$$\begin{cases} x_1^{(d)}(k+1) = a_{11}x_1^{(d)}(k) + a_{12}x_2^{(d)}(k) + a_{13}x_3^{(d)}(k) \\ x_2^{(d)}(k+1) = a_{21}p(k) + a_{22}x_2^{(d)}(k) + a_{23}x_3^{(d)}(k) \\ x_3^{(d)}(k+1) = a_{31}p(k) + a_{32}x_2^{(d)}(k) + a_{33}x_3^{(d)}(k) + \varepsilon \times \sin(\sigma \times p(k)) \end{cases}$$

In order to resist the divide-and-conquer attack on the self-synchronization chaotic stream cipher as shown above, the chaotic encryption algorithm p(k) should be designed as a nonlinear operator corresponding to the multiplication of two state variables $x_1^{(e)}(k)$, $x_2^{(e)}(k)$,given by

$$p(k) = \left(\operatorname{mod}(\left\lfloor \frac{\left\lfloor x_1^{(e)}(k) \times x_2^{(e)}(k) \right\rfloor}{2^{27}} \right\rfloor , 256 \right) \oplus m(k) = c(k) \oplus m(k)$$

where $\oplus$ denotes the bitwise exclusive OR operation, $m(3k - 2) = R(k)$, $m(3k - 1) = G(k)$, $m(3k) = B(k)$,$k = 1; 2; 3; \cdots$, and $R(k)$, $G(k)$, $B(k)$ denote the tri-color video signals.

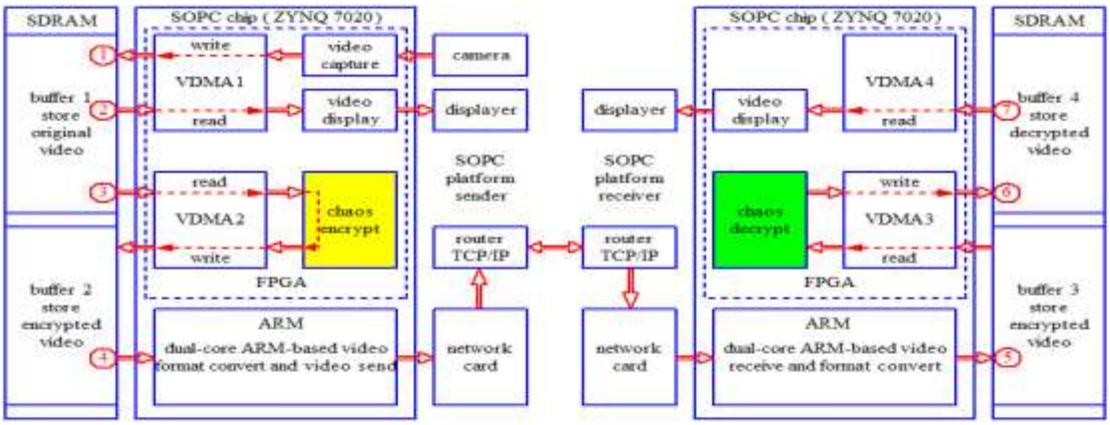


Fig. 2. Hardware implementation block diagram

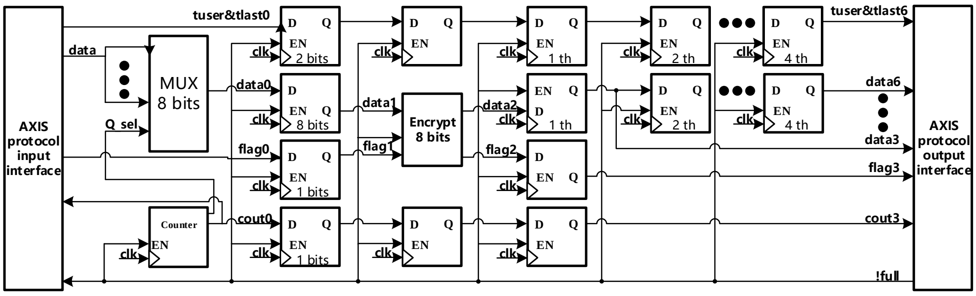


Fig. 3. Chaotic encryption modules

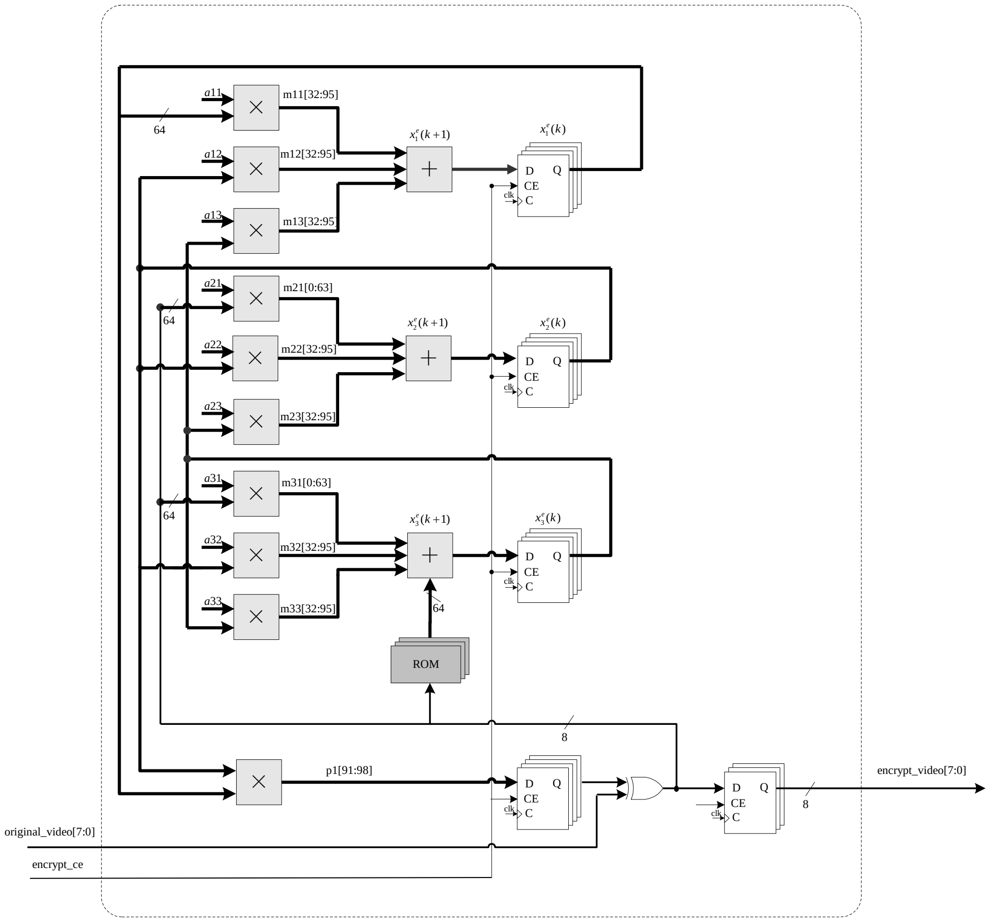


Fig. 4. Block diagram of FPGA design corresponding to the chaotic encryption algorithm

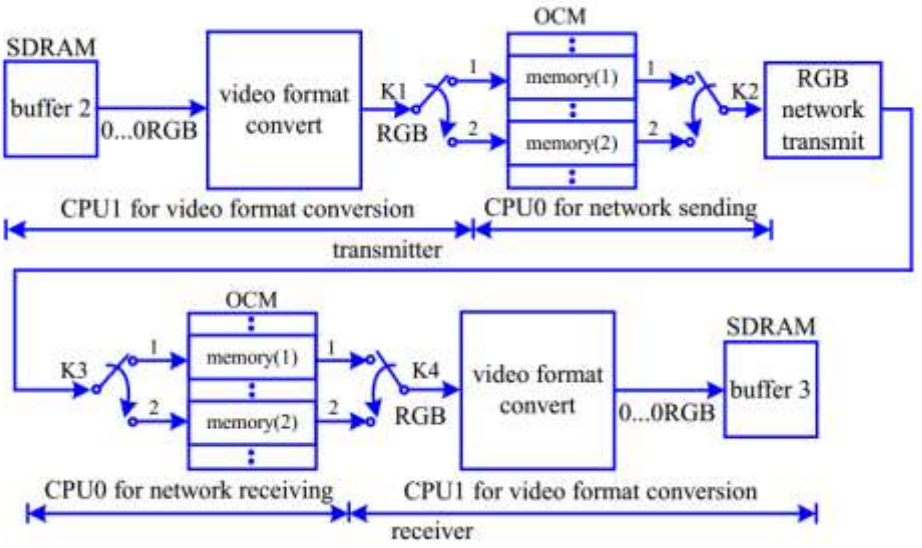
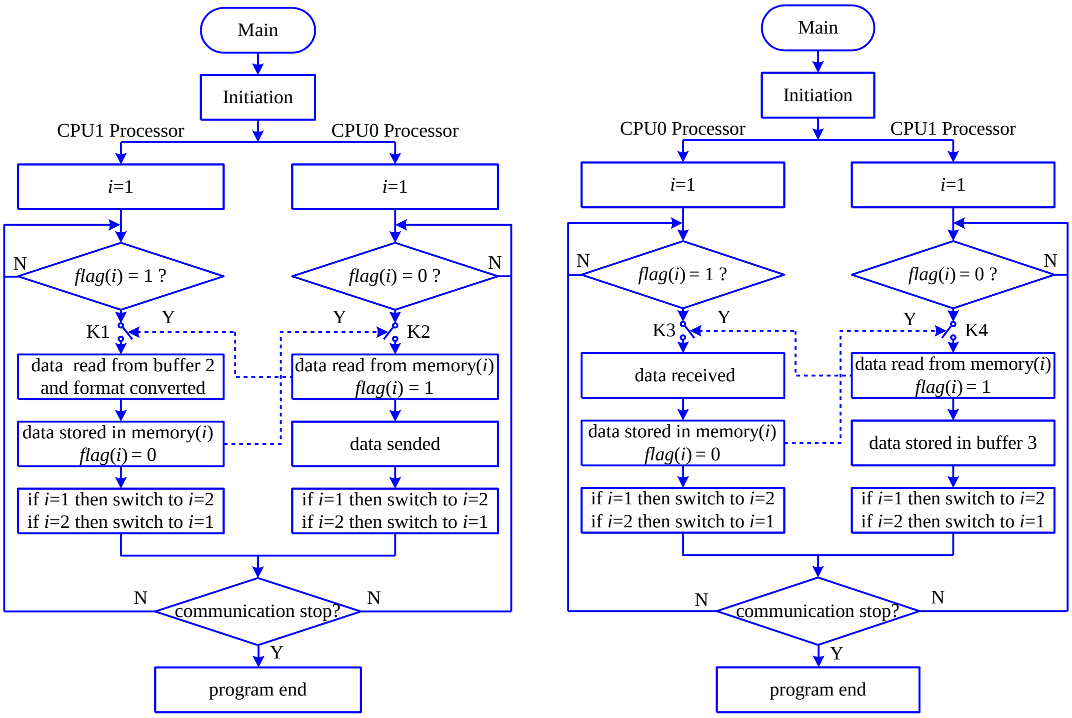


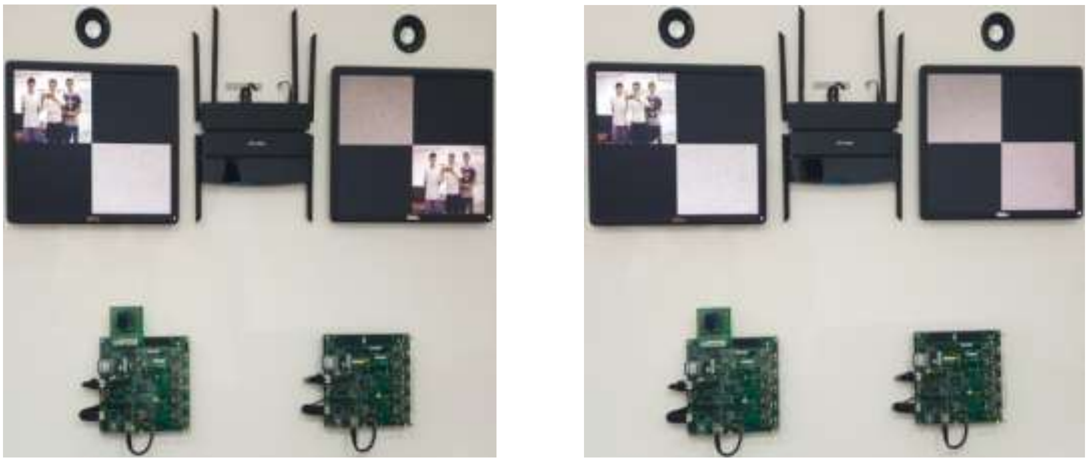
Fig. 5. Block diagram of dual-processor ARM-based video format conversion and transmission.


Main
Initiation
CPU1 Processor
CPU0 Processor
i=1
i=1
N
flag(i) = 1 ?
flag(i) = 0 ?
N
Y
Y
K1
K2
data read from buffer 2 and format converted
data read from memory(i) flag(i) = 1
data stored in memory(i) flag(i) = 0
data sended
if i=1 then switch to i=2 if i=2 then switch to i=1
if i=1 then switch to i=2 if i=2 then switch to i=1
N
communication stop?
N
Y
program end
Main
Initiation
CPU0 Processor
CPU1 Processor
i=1
i=1
N
flag(i) = 1 ?
flag(i) = 0 ?
N
Y
Y
K3
K4
data received
data read from memory(i) flag(i) = 1
data stored in memory(i) flag(i) = 0
data stored in buffer 3
if i=1 then switch to i=2 if i=2 then switch to i=1
if i=1 then switch to i=2 if i=2 then switch to i=1
N
communication stop?
N
Y
program end

(a) Software procedures at the transmitter.

(b) Software procedures at the receiver.

Fig. 6. Software procedures both at the transmitter and at the receiver



(a) Hardware experimental results with the matched keys

(b) Hardware experimental results with the mismatched keys

Fig. 7. Hardware experimental results(Resolution:640*480,FPS:31)