HARDWARE IMPLEMENTATION OF THE CODING ALGORITHM BASED ON FPGA

Margulan Ibraimov, Sakhybay Tynymbayev, Jongtae Park, Dauren Zhexebay and Madina Alimova

Almaty, Kazakhstan
The tasks

• to improve the performance of computing devices;
• to optimize the structure of the calculator;
• to use an efficient hardware device;
• to adapt the device of the coder and decoder depending on the bits of the input data;
• to implement a converting algorithm to visualize the original encrypted data.
RTL DIAGRAM OF THE MAIN PROGRAM BLOCK
Functional multiplier diagram of polynomials irreducible polynomials modulo
Table 1. Example of multiplying of polynomials modulo $R = A(x) \times B(x) \mod P(x)$,
$A(x) = x^4 + x + 1$; $B(x) = x^4 + x^2 + x + 1$; $P(x) = x^5 + x^3 + 1$

|   | $b_i$ | Partial reminder former | $b_i \times r_i$ | AD2          |
|---|------|-------------------------|------------------|--------------|
| START | $b_0 = 1$ | $r_0 = A = 10011$ | $b_0 \times r_0 = 10011$ | $R_0 = r_0 = 10011$ |
| TP1 | $b_1 = 1$ | $r_1 = 2r_0 \mod P(x)$ | $100110$ | $\oplus$ | $101001$ | $001111$ | $b_1 \times r_1 = 01111$ | $R_1 = r_1 \oplus R_0$ | $10011$ | $\oplus$ | $01111$ | $1100$ |
| TP2 | $b_2 = 1$ | $r_2 = 2r_1 \mod P(x)$ | $011110$ | $\oplus$ | $101001$ | $011110$ | $b_2 \times r_2 = 011110$ | $R_2 = r_2 \oplus R_1$ | $1100$ | $\oplus$ | $11110$ | $00010$ |
| TP3 | $b_3 = 0$ | $r_3 = 2r_2 \mod P(x)$ | $111100$ | $\oplus$ | $101001$ | $010101$ | $b_3 \times r_3 = 0$ | $R_3 = R_2$ | $= 00010$ |
| TP4 | $b_4 = 1$ | $r_4 = 2r_3 \mod P(x)$ | $101010$ | $\oplus$ | $101001$ | $000011$ | $b_4 \times r_4 = 00011$ | $R_4 = r_4 \oplus R_3$ | $00011$ | $\oplus$ | $00010$ | $00001$ |
DIVISION OF 32-BITDATA IN 15, 8, 5 AND 4-BITS

```verilog
class coder15 #(.W15, .W4) (data, A, clk, out)
begin
  .a(A),
  .clk(clk),
  .out(out15)
end
```

32 bit:  --> 4 bit
         --> 5 bit
         --> 8 bit
         --> 15 bit
The program for the formation of VGA sync signal and the return of the coordinates of the display pixel
Signal formation program (RGB) for each pixel
The timing diagram of the algorithm:

(a) $A(x) = x^4 + x^2 + 1$; $B(x) = x^4 + x + 1$; $P(x) = x^5 + x^3 + 1$,

(b) $A(x) = x^4 + x + 1$; $B(x) = x^4 + x^2 + x + 1$; $P(x) = x^5 + x^3 + 1$
The amount of resources spent for this algorithm on the Artix 7 FPGA

| Slice Logic Utilization | % of the FPGA resource used when encoding and decoding 4-bit code | % of the FPGA resource used when encoding and decoding 8-bit code | % of the FPGA resource used when encoding and decoding 12-bit code | % of the FPGA resource used when encoding and decoding 24-bit code |
|-------------------------|---------------------------------------------------------------|---------------------------------------------------------------|---------------------------------------------------------------|---------------------------------------------------------------|
| Number of Slice Registers | 0.02% | 0.1% | 0.39% | 1.25% |
| Number of Slice LUTs | 0.05% | 0.33% | 1.48% | 4.9% |
| Number of bonded IOBs | 7% | 13% | 19% | 36% |
| Number of BUFG/BUFGCT RLs | 3% | 3% | 3% | 3% |
CONCLUSIONS

Accordingly, possibility of establishing a fully autonomous encryption device of an n-bit digital code was shown. The main advantage of this method over other existing methods is that the algorithm compactly combines efficiency using embedded hardware modules for specific devices in the FPGA.

Accelerators the computation of cryptographic algorithms, particularly for cryptography based on a multiplier of polynomials irreducible polynomials modulo, to protect sensitive data within the framework of certain restrictions becomes significant in modern processor technology.
THANKS FOR YOUR ATTENTION!