Implementation of DES on Programmable Hardware

ABSTRACT

This paper discuses a hardware implementation of the Data Encryption Standard (DES) and its variants using FPGA technology. DES was implemented in Electronic Codebook Mode (CEB) mode and Cipher Feedback (CFB) mode, as well as a three-stage pipelined implementation of Triple Data Encryption Algorithm. It was found that the Altera CPLD FLEX10K device family was well suited to various aspects of the algorithm, especially the S-Boxes. The algorithm was modeled in VHDL and was synthesised for FLEX10K100 using the Altera Max Plus software. Furthermore the DES implementation was used for real-time secure voice communication using the CFB mode.

1. INTRODUCTION

The Data Encryption Standard [1,2] which also specifies the Triple Data Encryption Algorithm (TDEA), whilst currently being superceded by the new Advanced Encryption Standard, is nonetheless one of the most widely used means of private key data encryption worldwide. The DES and TDEA have applications in many areas of information security, which include data encryption for secure storage, secure voice communications, encryption of data for transmission, and data authentication.

The cryptographic algorithms are usually either implemented in software or in dedicated hardware. However, recent increases in the gate density of programmable devices makes it possible to implement sophisticated cryptographic algorithms on such devices. This not only enables major changes in the algorithm without replacing the hardware, but also provides increased security and better conformance to standards [3], with the entire algorithm being implemented on a single chip. Furthermore, the clockable speeds of such programmable devices are very high, with some device families rated as high as 200 MHz, thus allowing high throughput whilst retaining excellent reliability and performance.

2. DATA ENCRYPTION STANDARD (DES)

2.1DES Algorithm

The DES is a Feistel [4,5] transformation that operates on blocks of data. The algorithm is designed to encipher and decipher blocks of data consisting of 64 bits under control of a 64-bit key (every eighth bit of the key is discarded and the effective key length is 56 bits). Deciphering must be accomplished by using the same key as for enciphering, but with the schedule of addressing the key bits altered so that the deciphering process is the reverse of the enciphering process. Figure 1 illustrates the DES enciphering computation [4].

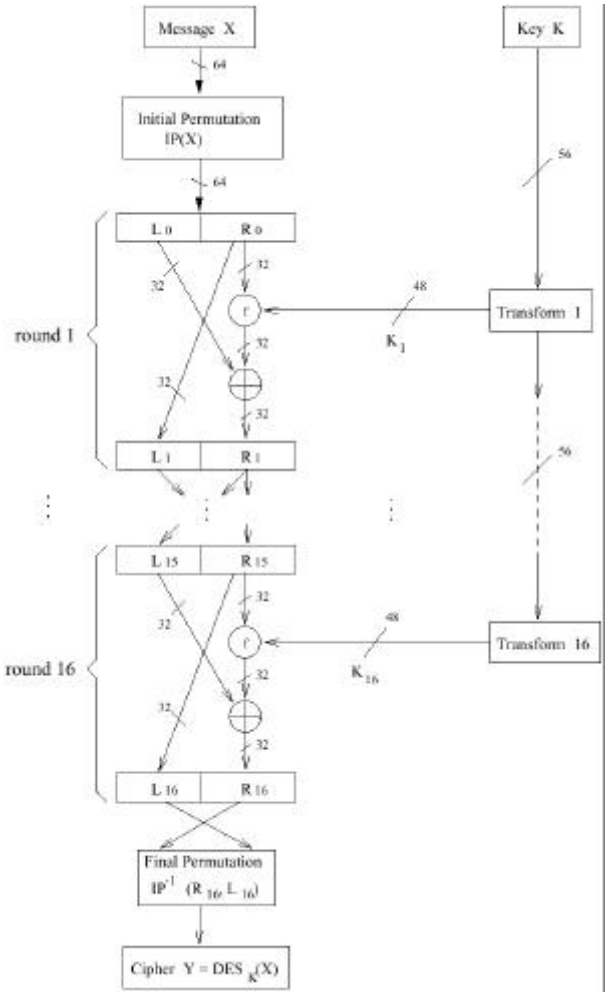

Figure 1: DES deciphering computation using Feistel Network

The operation of cipher function $f(R,K)$ in Figure 1 is demonstrated in Figure 2. The E in Figure 2 denotes a look-up table with 32 bit input and 48 bit output. Each of the unique selection functions $S_1,S_2,...,S_8$, takes a 6-bit block as input and yields a 4-bit result. The permutation function P yields a 32-bit output from a 32-bit input by permuting the bits of the input block.

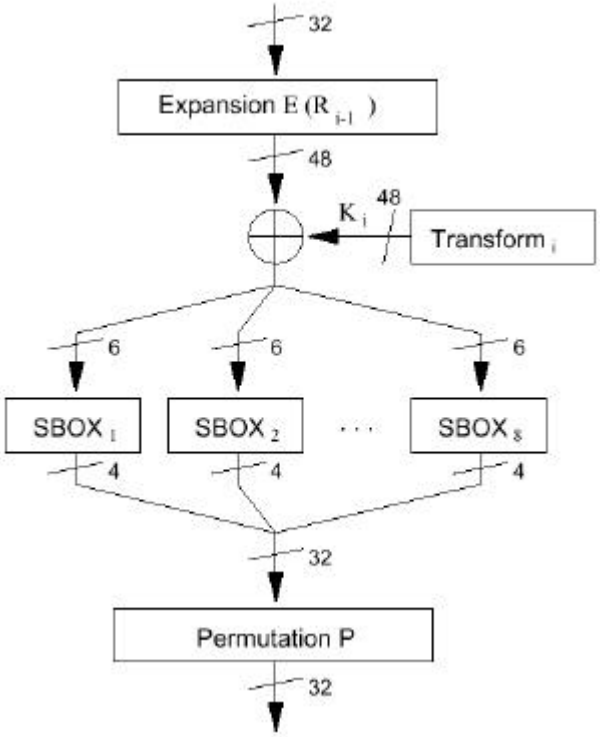

Figure 2: Computation of $f(R,K)$

Subkey K_n, for 1£n£ 16, is the block of 48 bits generated from *KEY*. That calculation is illustrated in Figure 3.

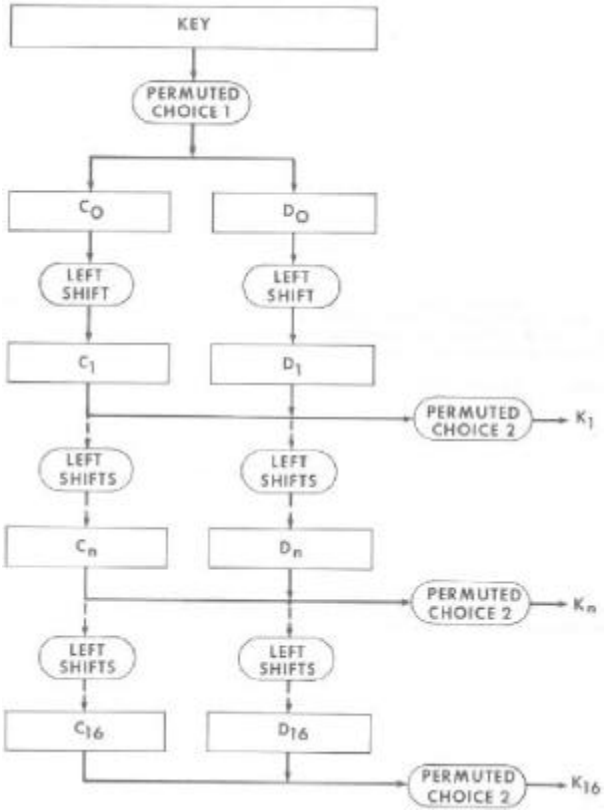

Figure 3: Sub key calculation

2.2 TDEA Algorithm

Triple DES or TDEA [1] is a DES based algorithm to provide more security as simple DES became more suspect. Let $E_K(I)$ and $D_K(I)$ represent the DES encryption and decryption of I using DES key K respectively. Each TDEA encryption/ decryption operation is a compound operation of DES encryption and decryption operations defined by $O = E_{k3}(D_{k2}(E_{k1}(I)))$ and $O = D_{k3}(E_{k2}(D_{k1}(I)))$ for encryption and decryption respectively. There are 3 key management options defined by the standard. K_1, K_2 and K_3 are independent, K_1, K_2 are independent, and $K_1 = K_3$, and $K_1 = K_3 = K_1 = K_3$, with the last option being same as DES, and is provided for back compatibility with DES.

2.3 DES Modes of Operation

The DES standard [5,6] specifies three modes of operation for DES and TDEA.

- *Electronic Codebook Mode* (*ECB*)
- *Cipher Block Chaining Mode* (*CBC*)
- *Cipher Feedback Mode* (*CFB*)
- *Output Feedback Mode* (*OFC*)

ECB mode is the simplest mode. The plaintext is divided into 64 bit long blocks X_i and each block is encrypted separately.

In CBC mode the plaintext message is divided into blocks of 64 bits. The first block is xored with a 64-bit Initialisation Vector (*IV*), and used at the input to the encryption algorithm. All subsequent blocks are xored with the encrypted previous block before they are encrypted: $Y_0 = e_k(X_0 \oplus IV)$ and $Y_i = e_k(X_i \oplus Y_{i-1})$ for $i \geq 1$. Figure 4 illustrates the operation of CBC

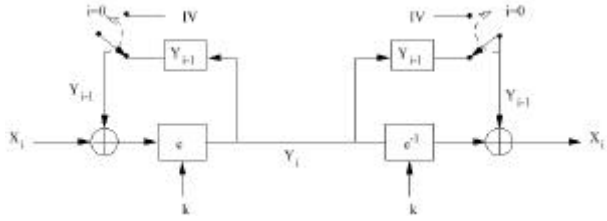

Figure 4: Cipher Block Chaining Mode

CFB mode is often used to encrypt messages smaller than 64 bits. The plaintext is divided into blocks of K bits ($1 \leq k \leq 64$). In the CFB mode of operation, an Initialization Vector (*IV*) of length l is used. The *IV* is placed in the least significant bits of the DES input block with the unused bits set to "0's" Figure 5 illustrates the operation of CFB.

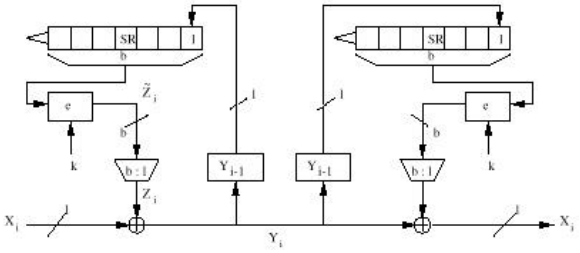

Figure 5: Cipher Feedback Mode

This input block is processed through the DES device in the encrypt state to produce an output block. During encryption, cipher text is produced by xoring a *k*-bit plaintext data unit with the most significant *k* bits of the output block. Similarly, during decryption, plain text is produced by xoring a *k*-bit unit of cipher text with the most significant *k* bits of the output block.

In both cases the unused bits of the DES output block are discarded. In both cases the next input block is created by discarding the most significant *K* bits of the previous input block, shifting the remaining bits *K* positions to the left and then inserting the *K* bits of cipher text just produced in the encryption operation or just used in the decryption operation into the least significant bit positions. This input block is then processed through the DES device in the encrypt state to produce the next output block.

OFB Mode is similar to CFB except that output of encryption function is used as feedback and not the ciphertext.

3. DESIGN ARCHITECTURE

3.1 DES Implementation in ECB Mode

The design of DES was implemented in two major VHDL modules. One module implemented the controlling FSM and the *L* and *R* registers in Figure 1, and the other module implemented the function *f*(*R, K*), calculation and the 16 sub-key generation in Figure 2 and 3 respectively. In addition the design was flexible enough to switch from encryption to decryption on a block by block basis on the fly.

The crucial part of the design was the implementation of the eight S-boxes in the *f*(*R, K*) function. They used ROM in the Embedded Array Blocks (EAB) in the FLEX10K100 chip [7].

The permutation and shifts were implemented using wiring resources only (no logic required).

The function sub-keys were all generated simultaneously initially. Therefore it was possible to use the same input pins both for key and the data.

3.2 TDEA Implementation

Since the FLEX10K100 had 12 EABs it was possible to implement each of encryption/encryption stages in pipelined fashion. However, in this design each EAB implements two S-boxes. The clock signal was used as the most significant bit of the address to the ROM, such that when the clock was high, the first of the two S-boxes was selected, and when the clock was low, the second S-box was selected. In this way, during the course of a complete clock cycle, both S-boxes would have contributed to the output. A clocked demultiplexor arrangement on the output can be used to concatenate the four-bit slices. This clearly places a limit on the maximum clock speed.

3.3 DES Implementation in CFB mode

Implementation of the CFB mode required some additional hardware, including a multiplexer, a shift register, and an extra *k* XOR gates (for a k-bit implementation of CFB mode DES). Both k =8 and k = 1 versions were implemented. Because in *k*-bit CFB mode, the final $64 - k$ bits of the output from IP^{-1} are discarded, it is unnecessary to store the entire permutation.

3.4 Secure Voice Communication

DES in CFB mode with k = 8 was used for secure voice communication. The voice was digitized using an 8-bit ADC, and converted back to digital data using a DAC. Buffer amplifiers were employed before the ADC and DAC stages. In addition anti-aliasing filters on the input and low-pass filter on the output stage were employed to attenuate the quantisation and other induced noises.

4. RESULTS

The maximum clock frequency for the implementation in ECB mode was 24.33 MHz. This corresponds to a data rate of 98 Mbits/s.

The clock frequency for the TDEA Implementation was 19.84 MHz, which corresponds to a data rate of 80 Mbits/s.

The 8-bit and 1-bit CFB implementations achieve clock frequencies of 32.15 and 30.39 MHz respectively.

4. CONCLUSION

Commonly used encryption algorithms can now be implemented in FPGA chip sets, and achieve respectable data rates. A crucial part of the resources in such implementations is ROM to implement substitution boxes. This is borne out by other work on the AES (Rijndael) where the implementation of its s-boxes consumes most of the chip real estate, and determines the overall data throughput rates

REFERENCES

[1] Federal Information Standard Publications, FIPS 46-3, "*Data Encryption Standard"*, US Department of Commerce, 1999.

[2] ANSI X3.106, "*American National Standard for Information Systems – Data Encryption Algorithm – Modes of Operation*", American Standards Institute, 1983.

[3] Federal Information Standard Publications, FIPS 140-1, "*Security Requirement for Cryptographic Modules*", US Department of Commerce, 1980.
[4] B. Schneier, "*Applied Cryptography: Protocols, Algorithms, and Source Codes, in C*", Wiley, 2nd Ed, 1995.
[5] S. A. Vanstone, A.J. Menezes, and P.C. Van Ooschot, "*Handbook of Applied Cryptography*", Discrete Mathematics and its Applications, CRC Press, 1997.
[6] Federal Information Standard Publications, FIPS 46-81, "*DES Modes of Operations*", US Department of Commerce, 1980.
[7] Altera Corporation, "*Data Book*", 1998.

<Insert reference to Nicholas's thesis>

<Suggest Nooshabadi S, Rowcroft N and Sale AHJ as the author order because you need the first place and Nick did the work. Besides, it is alphabetical in last name.>

<BTW, The Enigma project was far better done!>