Hardware Evaluation of the AES Finalists

Tetsuya ICHIKAWA* Tomomi KASUYA** Mitsuru MATSUI**

* Kamakura Office, Mitsubishi Electric Engineering Company Limited
ichikawa@harriet.mee-unet.ocn.ne.jp
** Information Technology R&D Center, Mitsubishi Electric Corporation
kasuya@iss.isl.melco.co.jp, matsui@iss.isl.melco.co.jp

1. Introduction

This report describes our evaluation results of implementing hardware of the AES finalists, concentrating on 128-bit key version, using Mitsubishi Electric's 0.35 micron CMOS ASIC design library. Our goal is to estimate the "critical path length" of data encryption /decryption logic and key setup time of key scheduling logic for each algorithm, which corresponds to the fastest possible encryption speed in feedback modes of operation such as CBC etc. To achieve this, we wrote fully loop-unrolled codes in Verilog-HDL language without introducing pipeline structure that blocks the feedback.

We first tried to investigate the evaluation environments to be used in NSA, especially the hardware design library, since NSA is expected to join the Round Two hardware analysis as has been shown in the NIST AES homepage [NIST (1998)]. However, after communicating NIST and MOSIS, we found that the library is an internal 0.5 micron standard cell library that is not available outside NSA, and a non-proprietary version of the library has not been developed. We therefore decided to analyze the AES finalists using Mitsubishi Electric's CMOS ASIC design library, whose information is publicly available in [MITSUBISHI (1997)].

Our simulation results show that Rjindael is the fastest as expected and it is even faster than DES, and Serpent is the next. Twofish, Mars and RC6 are slower than Triple-DES. We should note that since we used a general ECA (embedded cell array) library without applying special performance optimization techniques, these algorithms that heavily use arithmetic operations could be much faster if we introduce more expensive semi- or full-custom designs. However our analysis also indicates that even such designs are not expected to give a significant impact to change the ranking of the critical path length.

2. The AES Finalists

NIST announced the five AES finalists, in August 1999. This section briefly summarizes these algorithms, mainly data encryption operations, from hardware viewpoint.

2.1 Mars

Mars supports 128-bit blocks and a variable key size from 128 bits to 448 bits. It is designed to take advantage of the powerful operations supported on today's computers [Burwick et. al. (1999)].

The encryption part of Mars, which is composed of four kinds of round functions, is performed as follows. We have also listed major components that have an impact in hardware performance.

-The initial key addition
  4 additions mod 2^{32}.
-The unkeyed forward mixing (8 rounds)
  2 additions mod 2^{32}, and 4 look-up tables
  with 8bit-input/32bit-output.
-The keyed forward transformation (8 rounds)
  6 additions and 2 multiplications mod 2^{32},
  and 4 data-dependent rotations.
-The keyed backwards transformation (8 rounds)

6 additions and 2 multiplications mod 2^{32},
and 4 data-dependent rotations.
-The unkeyed backwards mixing (8 rounds)
2 subtractions mod 2^{32}, and 4 look-up
tables with 8bit-input/32bit-output.
-The final key addition
4 subtractions mod 2^{32}.

It seems that the heavy use of arithmetic operations, especially multiplications and additions mod 2^{32}, makes hardware slower and larger unless they are specially designed in a transistor level.

2.2 RC6

RC6 has three variable parameters, i.e., the number of rounds, the data block size, and the key size up to 2040 bits. The proposed version in AES has 20 rounds with a total of 4 additions (subtractions) mod 2^{32} before and after the round functions [Rivest (1998)], [RSA (1998)]. The major hardware components in the round function are as follows:

2 additions and 2 multiplications mod 2^{32},
2 data-dependent rotations.

These operations are well supported and fast on modern microprocessors, but expensive in hardware, especially multiplications and additions mod 2^{32}, make hardware slower and larger unless they are specially designed in a transistor level.

2.3 Rijndael

Rijndael also has a variable block length and a variable key length. The block length and the key length can be independently specified to 128, 192 or 256 bits. The proposed number of rounds in AES is 10, 12 and 14 when the key length is 128 bits, 192 bits and 256 bits, respectively [J.Daemen and V.Rijmen (1998)].
The round function of Rijndael in 128-bit blocks is composed of four distinct invertible transformations as follows:

-The ByteSub transformation
16 lookup tables with 8bit-input/output.
-The ShiftRow transformation
no hardware operations.
-The MixColumn transformation
logical AND and XOR operations.
-The AddRoundKey transformation
logical XOR operations.

Before the first round, the AddRoundKey transformation is also performed, and in the final round, the MixColumn transformation is omitted.
The basic components of Rijndael are logical operations and lookup tables; the latter is actually a composite function of an inversion over GF(2^8) with an affine mapping. Hence the structure of Rijndael is expected to be suitable for hardware implementation.

2.4 Serpent

Serpent has a 32-round SP-network structure with initial and final permutations, whose round function consists of 32 lookup tables with 4-bit input/output, logical and rotate shifts, and XOR operations [Anderson, Biham and Knudsen (1998)], [Biham (1997)].
These components are suitable for hardware implementation; particularly the small table size is expected to make hardware sufficiently small and fast.

2.5 Twofish

Twofish has a 16-round Feistel-like structure with an additional whitening of the input and output that consists of XOR operations. The major hardware components of the round function are as follows:

n lookup tables with 8-bit input/ output,
4 additions mod 2^{32},
logical AND and XOR operations,

The lookup tables can be also generated from another smaller 8 lookup tables with 4-bit input/output, and ***n*** is 12, 16 or 20 when the key length is 128, 192 and 256, respectively.
Twofish is not using particularly heavy operations in hardware, but its critical path is not short because, for instance, the number of cascaded 8x8 lookup tables is 48, where that for Rijndael is 10 when the key length is 128 [B.Schneier et. al. (1998)].

3. Design Policy

Our purpose is to evaluate the fastest possible encryption speed of the AES finalists using the existing hardware library under fair conditions. To achieve this and also to complete the analysis in our limited time scale and resources, we designed the 128-bit key version for each candidate on the basis of the following criteria and conditions:

1. We fully unrolled the loop in the encryption and decryption logic and the key scheduling logic to achieve the fastest possible speed (throughput). In practice, the loop structure is commonly used in order to reduce hardware size, but generally makes the hardware slower because additional setup -time and hold-time is required for the loop registers, which is usually not negligible. Note that we therefore did not take a special effort to reduce hardware size.

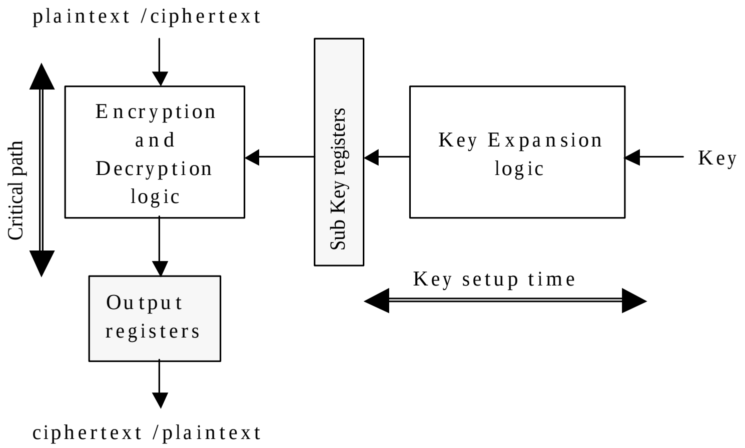


Figure 3.1 The hardware structure

2. We assume that all subkey bits are stored in subkey registers before an encryption operation begins. Also we have inserted another 128-bit resister to hold a block of ciphertext as shown in Figure 3.1, where we define the critical encryption and decryption path as the time required for all output bits of the encryption and decryption logic to reach the output registers under the fixed (sub)key value.

3. We did not introduce pipeline architecture; i.e., we did not insert any additional intermediate registers in the encryption and decryption logic. This is because the pipeline architecture makes the ECB mode faster but also blocks feedback modes of operations such as CBC. In other words, our hardware model encrypts one block plaintext data in one cycle.

4. We did not use a special optimization technique to design lookup tables in hardware. This means that the performance of the lookup tables heavily depends on optimization capability of the logic synthesis tool. In practice, as will be shown in the next section, the output of the synthesis tool seems to have reasonably optimized the lookup tables (not very slow).

5. Our design environment is as follows:

language:	Verilog-HDL
simulator:	Verilog-XL
design library:	Mitsubishi 0.35micron CMOS ASIC Library
logic synthesis:	Synopsys Design Compiler version 1998.08

For arithmetic operations such as additions, subtractions and multiplications, we used faster ones in the library of Synopsys Design Ware Basic Library [Synopsys (1998)]. Also, we adopted the WORST case hardware conditions for evaluation. The worst case speed is a guaranteed speed of a given circuit, which is commonly used in real products. We think that the TYPICAL case evaluation is too optimistic to apply to a real ASIC hardware.

4. Evaluation Results

The results of our hardware evaluation of the five finalists are presented in Table 4.1. The fastest algorithm in terms of the critical path between plaintext and ciphertext is Rijndael, which is an only algorithm faster than DES. The second fastest algorithm is Serpent, which is twice faster than triple-DES but still much slower than Rijndael (approximately half). The speed of Twofish is almost the same as that of triple-DES, but Mars and RC6 are further slower; Rijndael is approximately ten times faster than RC6.

On the other hand, for the key setup time, Twofish is fastest, consuming only 5% of the critical path of its encryption procedure. Note however that the key setup time of DES and Triple-DES is almost nothing in hardware. Rijndael and Serpent have approximately 85%, while the key scheduling logic of Mars and RC6 is more than three times slower than their encryption.

Figures 4.1 and 4.2 show more detailed breakdowns of hardware components on the critical path of each algorithm, where the horizontal line of Figure 4.2 is normalized to show proportion of each component .

Mars has 16 multiplications, 26 additions/ subtractions, 15 lookup tables (specifically 11 S0's and 4 S1's) and 9 data-dependent rotations on its critical path, where all arithmetic operations are taken on mod 2^{32}. As shown in the figures, the multiplications occupy 63% of the critical path, 13% for additions/subtractions, and 9% for the lookup tables.

RC6 has 20 multiplications, 21 additions and 20 data-dependent rotations on its critical path, where all arithmetic operations are also taken on mod 2^{32} As shown in the figures, the multiplications occupy 77% of the critical path, 13% for additions/subtractions, and 8% for the data dependent rotations.

The critical path of Rijndael is not in the encryption but in the decryption procedure since the InvMixColumn function, which is an inverse of the MixColumn function, is a bit slower than the MixColumn function due to more complex constant values. On the critical path, a total of 10 InvByteSub functions (table lookups) occupy 48% of the entire decryption time, and a total of 9 InvMixColumn functions have 43%.

It is easy to see that the critical path of Serpent has 32 lookup tables and 31 linear transformations (XOR's and shifts). Our analysis shows that the linear transformations of Serpent are more expensive than its lookup tables; the former is 36% while the latter is 45%. In a logical sense, the lookup tables and the linear transformations must exhaust the critical path; however Figure 4.2 exhibits other factors that occupy a total of 19%. This is mainly because the design compiler has automatically inserted driver gates in order to supply sufficient fan-out counts, which reflects the fact that an output bit of a lookup table of Serpent has many “branches” that reach many different lookup tables in the next round. This is part of design criteria of Serpent.

It is also easily seen that the critical path of Twofish have 48 lookup tables --- specifically 16 q0's and 32 q1's, which is not a trivial fact ---, 16 MDS's (linear transformations) and 32 additions mod 2^{32}. The dominant part is the lookup tables, which occupy 53%, but also time for additions is not negligible (28%).

5. Discussions and Conclusions

The performance of Mars and RC6 heavily depends on the speed of the multiplication circuits mod 2^{32}. Our evaluation results show that the average time for the multiplication is around 23ns, which is six to eight times slower than the addition circuit mod 2^{32}, which takes around 3ns.

This also shows that by using highly optimized multiplication circuits in a transistor level, these algorithms are expected to be much faster. For this topic, see [Hagi (1998)] for instance. Now as an example, let us assume, in Mars and RC6, the 32-bit multiplication can work at the same speed as the 32-bit addition. We see that still the critical path of (the modified) Mars and RC6 is approximately 250 and 200ns, respectively. Also, we should notice that a full-custom solution is generally process-dependent and hence is not an inexpensive solution in practice.

Another speeding-up possibility is to optimize a lookup table. The average time for one lookup table for each algorithm is 3.2ns for Rijndael (8x8), 1.5ns for Serpent (4x4), 3.5ns for Twofish (8x8) and 3.5ns for Mars (8x32), respectively. Twofish will be most rewarded for the efforts of optimizing the lookup tables. However, the optimization will not lead to a significant impact to affect the ranking of the five finalists.

In this paper, we did not take efforts to reduce the size (area) of each algorithm since we adopted a full loop unrolling in order to evaluate the fastest possible encryption speed. Appendices 1 and 2 show the information of the size of each algorithm with the detailed breakdowns, which we will not discuss here. How to reduce the gate size is another practical topic to be pursued.

References

[NIST (1998)]:
http://csrc.nist.gov/encryption/aes/aes_home.htm

[MITSUBISHI (1997)]:
Mitsubishi Electric America, Inc.,
"0.35um CMOS ASIC DATA BOOK ", 1997.

[Burwick et.al. (1999)]:
http://www.research.ibm.com/security/Mars.html
See also
http://csrc.nist.gov/encryption/aes/round2/AESAlgs/MARS/mars-int.pdf

[Rivest (1998)]:
R. L. Rivest, M. J. B. Robshaw, R. Sidney, and Y. L. Yin, "The RC6 Block Cipher," 1998.

[RSA(1998)]:
http://www.rsasecurity.com/rsalabs/aes/index.html

[J.Daemen and V.Rijmen (1998)]:
J. Daemen and V. Rijmen, "AES Proposal: Rijndael," Document vers on 2, Date: 03/09/99.
http://www.esat.kuleuven.ac.be/~rijmen/rijndael

[Biham (1997)]:
E Biham, "A Fast New DES Implementation in Software", in Fast Software Encryption - 4th International Workshop, FSE '97, Springer LNCS v 1267,pp 260-271.

[Anderson, Biham and Knudsen (1998)]:
R. Anderson, E. Biham and L. R. Knudsen, "Serpent: A Proposal for the Advanced Encryption Standard," 1998.
http://www.cl.cam.ac.uk/~rja14/serpent.html

[B.Schneier et. al. (1998)]:
B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall, and N. Fergusen, "Twofish: A 128-Bit Block Cipher," June 15, 1998.
http://www.counterpane.com/twofish.ps.zip

[Synopsys (1998)]:
Synopsys Inc. ,"Design Ware Foundation Quick Reference Guide ", Aug.1998.

[Hagi (1998)]:
Y.Hagihara, et. al., "A 2.7ns 0.25um CMOS 54x54b Multiplier", ISSCC Digest of Tech. Papers, pp296-297, Feb.1998.

Table4.1 Hardware evaluation results

Algorithm name	area [Gate]			Key setup time[ns]	Critical-path[ns]	Throughput [Mbps]
	Encryption & Decryption	Key Schedule	Total			
DES	42,204	12,201	54,405	-	55.11	1161.31
Triple-DES	124,888	23,207	148,147	-	157.09	407.4
MARS	690,654	2,245,096	2,935,754	1740.99	567.49	225.55
RC6	741,641	901,382	1,643,037	2112.26	627.57	203.96
Rijndael	518,508	93,708	612,834	57.39	65.64	1950.03
Serpent	298,533	205,096	503,770	114.07	137.4	931.58
Twofish	200,165	231,682	431,857	16.38	324.8	394.08

Figure 4.1 Critical Path of the Finalists(1)

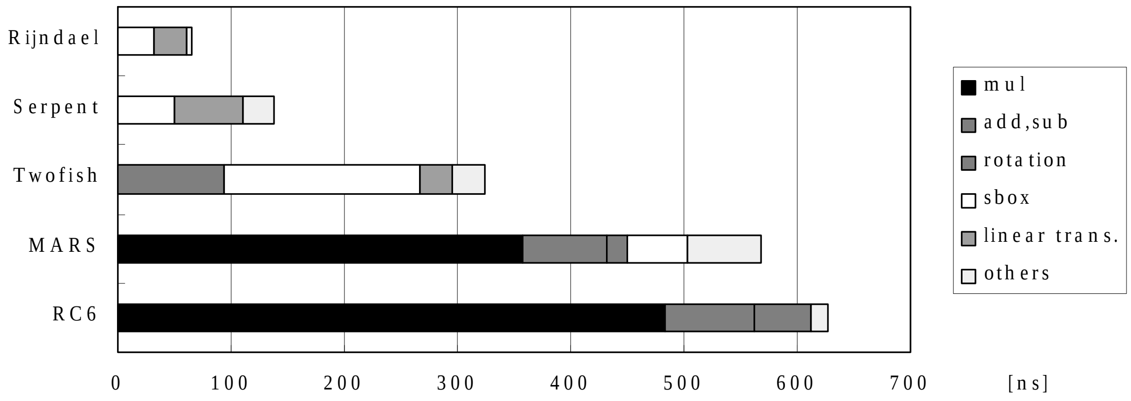
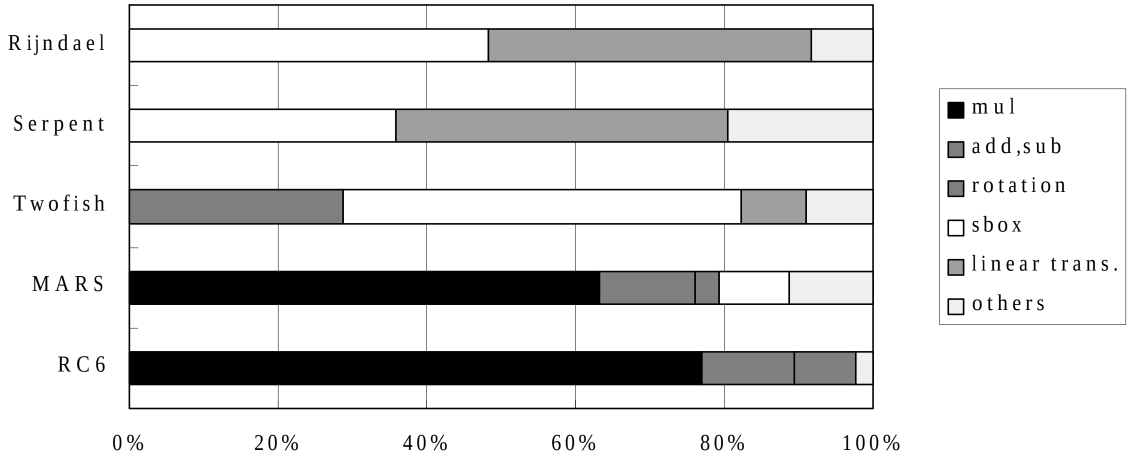
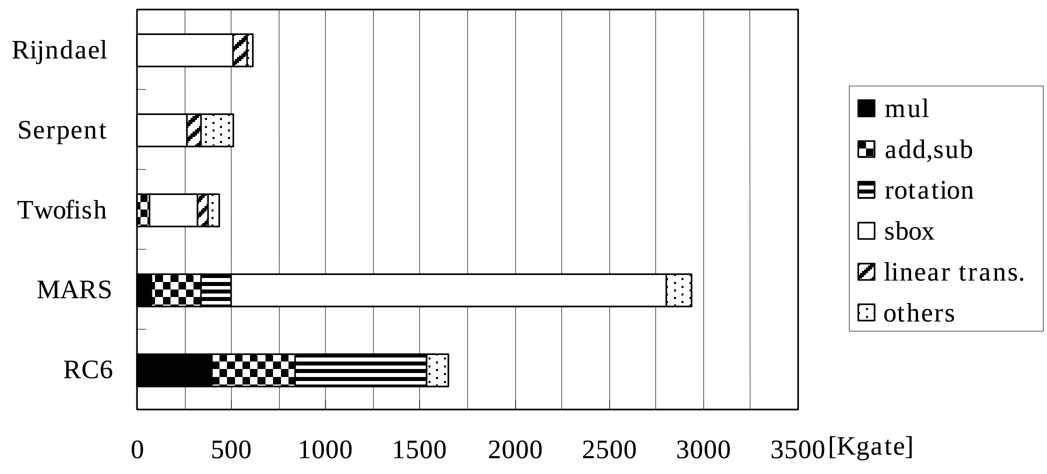


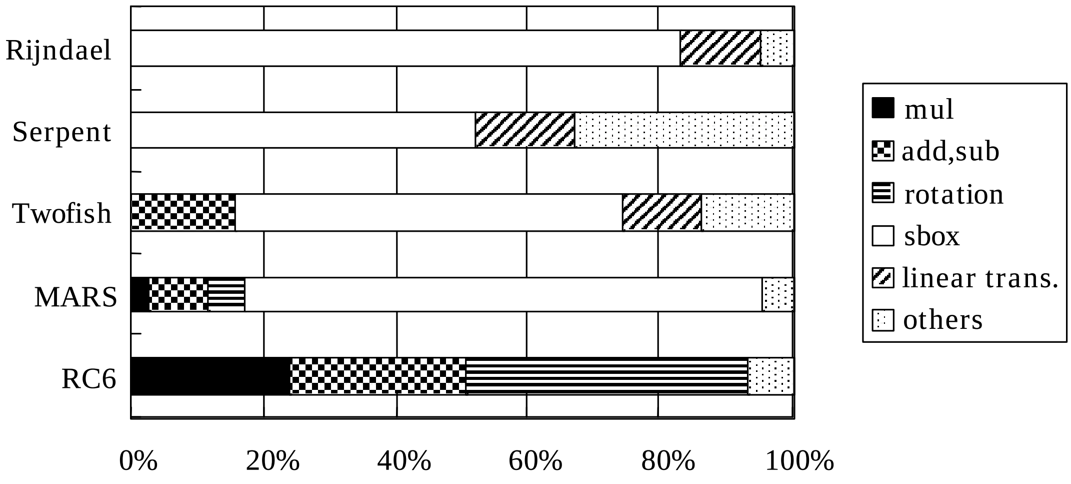
Figure 4.2 Critical Path of the Finalists(2)



Appendix 1: Area Size of the Finalists(1)



Appendix 2: area size of the Finalists(2)


Rijndael
Serpent
Twofish
MARS
RC6
mul
add,sub
rotation
sbox
linear trans.
others