通信情報保護學會誌
第2卷, 第2號, 1992. 6

음성정보의 공개열쇠방식 암호화를 위한 반도체 공정기술평가†

Evaluation of CMOS process for public key encryption of telephone service

한선경 · 유영갑*

요 약

전화망을 통과하는 음성신호에 대하여, 실시간에 공개열쇠방식의 암호화/복호화를 하기 위한 반도체 IC 제조공정평가를 실시하였다. 초당 64k bit의 정보에 대하여 256 bit 이상의 key를 갖는 RSA 방식 암호화를 위하여 modular multiplication 환경과 redundant number system을 채택하여 algorithm 상에서 최고속 계산이 이루어지는 것이 가정되었다. 이 최고속 algorithm에 대한 speed 요구를 multiple input shift register를 사용하는 회로로 충족시키는 과정에서, 1.0μ 이하의 CMOS 공정이 요구된다는 결론에 도달하였으며, 이들 회로의 타당성은 저속 RSA chip의 분석 결과와 비교하여 확인하였다.

1. 암호통신구현의 의의

정보의 경제적 가치는 통신망의 확충과 함께 크게 증폭되고 있다. 특히 정보의 수요가 다양화하는 추세는 전기통신망의 급격한 보급에 힘입어 더욱 가속화하고 있다. 정보 교류의 양적 팽창은 보다 광범위하고 효율적인 통신망의 수요를 낳고 있으며, 따라서 보다 빠르고, 대용량의 고품질의 정보 서비스 체제의 등장을 요구하고 있는 것이다.

정보 서비스의 고 품질화에서 정보가치를 극대화 시키려는 여러가지 노력이 있어 왔다. 여기에는 수집 또는 생산된 정보의 정확성과 시간적 유효성을 유지시키려는 노력과 함께 이 정보를 지정된 수요자들에게만 전달하는 작업이 포함되어 있다. 정보가치의 창조 또는 향상과 함께 정보가치의 유지가 중요한 문제가 되고 있는 것이다. 정보를 이용하는 경제활동의 요체는 정보의 유통체제 확립인데, 대부분 개방된 통신망 환경하에서 이 정보가치의 유지는 적지않은 비용발생을 유발하고 있는 것이다. 이것은 특정정보 유통을 위한 별도의 통신체제 구축과 고도의 암호화 기법의 본격적인 채택을 요구하고 있다. 이 연구는 정보가치의 주요 유지 수단인 암호체제구축에 필수적인 암호용 집적회로의 개발에 관한 것이다.

최근의 공중전화망의 다양한 활용은 음성통신분야의 정보보호의 수요를 제기하고 있다. 특히 근거리 무선전화기의 가입자간 혼선과 오접속등에 의한 통화

† 이 연구는 체신부, 한국통신의 지원으로 수행하였습니다.
* 충북대학교 공과대학 정보통신공학과

내용의 노출은 시급히 보호해야 할 필요가 있다. 여기에 덧붙여서 이동통신망의 급속한 확충은 향후의 전화망의 근본개념에 재정립을 요구하고 있으며, 노출된 무선신호를 이용하는 음성정보 등에 대한 적극적인 보호 없이는 심각한 혼란이 예상되는 것이다. 여기에서 정보노출에 대한 바람직한 해결책으로서 고려되고 있는 것이 암호화이다. 지금까지의 공중전화망에 대한 정보보호는 관용암호체제를 이용하여 왔기 때문에 정보보호의 핵심이 되는 암호화 열쇠의 분배와 관리의 문제가 암호서비스의 광범위한 보급에 장애 요인이 되는 것이다. 따라서 열쇠의 관리가 비교적 용이한 공개 열쇠암호체제가 방대한 숫자의 공중전화망 가입자의 정보보호에 현재로서는 가장 적합한 방식인 것이다. 이 방식의 구현은 궁극적으로 가장 값 싼 공중통신망의 실시간 음성서비스분야의 정보보호인 것이다.

그러나 공개열쇠 암호체제의 암호화와 복호화 과정에는 막대한 계산부담이 따르기 때문에, 이를 실시간 암호체제에 도입하는 것은 구현방법상 획기적인 개선을 요구하고 있다. 이 요구에 맞추어서 긴 자리수에 대한 지수계산시간을 단축시키기 위하여, 특별한 연산 알고리즘과 숫자표현 방식 그리고 전자회로 설계 등의 개선에 많은 진전을 가져왔다. 또한 지수 계산과정에서 계산의 대상이 되는 숫자의 표현에 redundant number system을 도입하므로써, 회로설계에서 필요한 기본 지수계산의 시간단위의 정량화가 가능하게 되었다. 이런 진전에도 불구하고 이 방식의 실질적인 구현의 장애는 이들 회로를 구성하는 기본 소자의 동작속도가 음성주파수의 데이타 발생비율에 맞출 수 없다는 제약이었다.

이 연구는 급격하게 개선되고 있는 반도체 소자의 동작특성을 공개열쇠방식의 암호화 및 복호화 계산에 적용할 수 있는 지의 여부를 정량적으로 평가하여 그 결과를 암호칩의 설계에 적용하는 것이다. 정보보호 체제중에서 가장 주목을 받고 있는 공개 열쇠방식의 구현의 문제를 다루고 있다. 특히 반도체 설계 및 제조기술이 submicron의 범위로 작아지게 되고, 이에 따라 회로 동작속도가 현저하게 빨라지게 되었다. 암호수요는 종래의 비 실시간 데이타 전송에서 실시간 음성신호에까지 확대될 것이고, 반도체 기술을 이용하여 효과적으로 구현할 수 있는 가능성을 증명해 보이고자 하는 것이다. 이를 위하여 문헌조사 및 분석, 기존의 암호칩의 분석, 암호화회로의 설계 및 simulation 그리고 완성된 칩의 실장환경 구축에 대하여 심도있는 연구개발 활동을 수행하였다.

특히 RSA 암호방식의 가장 큰 문제점인 지수계산의 시간부담을 줄이기 위하여, 일찍부터 modular 곱셈이 채택되고 있고 상당한 효과가 있는 것으로 파악되었다. 여기에 숫자표기에 있어서 연산 중간 값들에 redundant number system을 과감히 도입함으로써 연산 과정에 대한 계산 부담을 시간적으로 정량화할 수 있는 획기적인 길이 열리게 되어, 음성신호의 실시간 처리가 가능하게 되었다. 또한 RSA 암호방식에 관한 암호 시스템과 IC 제조업체의 data sheet나 기술문서등을 수집 분석하여 chip configuration상에서 기존 시스템과의 호환성을 확립할 수 있도록 하였다.

설계기간의 단축과 간접적인 실증 데이타 확보를 위하여 기존의 RSA 암호방식을 채택한 chip 분석을 실시하였다. chip의 분석과정은 decap 작업, passivation 제거작업, metal층과 적층구조 제거작업을 수행하였고 그 결과물에 대하여 3600배의 사진촬영을 실시하였다. 이 사진들을 이용하여 chip의 mosaic의 형태의 대형 사진으로 완성되어 여기에 대하여 회로추출이 실시되었다. 이 분석결과는 이론적인 연산 logic의 구조와 비교, 분석되었으며, 실제구현의 미세부분에서의 최적화 과정을 파악하는데 결정적인 단서가 되었다.

암호회로의 설계는 주로 지수계산의 시간부담을 줄이기 위한 fast exponentiation과 redundant number system을 이용하는 adder의 comparator logic에 대하여 simulation과 공정요구사항 정립에 주력하였다. 이것은 구체설계과정에 들어가기 전에 충분한 설계환경을 구축하기 위하여 반도체 설계 및 제조공정상의 최소선폭 변동에 의한 속도평가가 우선되어져야 하기 때문이다. 이것을 이용하여 제조된 chip의 성공적인 동작, 최소한의 기능적 동작이 보장되기 때문이다. 이 결과 얻어진 것은 음성주파수의 암호서비스를 위하여는 최소한 1.0μ 이하의 선폭을 갖는 CMOS 공정이 필요하다는 것이다. 이 논문의 제 2 장에서는 실질적인 구현이론을 3장에서는 목표소자의

설계과정을 다루었다.

2. RSA 방식구현이론

공개열쇠방식의 근간은 큰 숫자에 대한 지수계산을 통하여 필요한 암호문을 얻으므로써, 열쇠없이는 해독이 지극히 어렵도록 하는 것이다. 그러나 암호화에 사용되는 큰 숫자에 대한 지수계산은 심한 계산부담을 주게 되고, 이를 직접 음성신호대역에 적용하는 것은 현실적으로 큰 어려움을 야기시키는 것이다. 여기서는 이 지수계산과정을 신속하게 수행하기 위한 숫자표현방법, 계산 알고리즘을 설명하고자 한다.

가. Modular 곱셈방법

RSA 암호화 과정에서 사용되는 modular 곱셈은 기억용량의 급격한 증가와 과도한 계산시간의 문제점이 있다. 평문을 숫자화하여 여기에 큰 지수를 써서 지수계산이 수행되고 나면 대단히 큰 자릿수의 숫자가 얻어지게 된다. 즉 n 자리의 숫자를 m 승하게 되면 n * m 자리의 숫자가 얻어지게 되고, 이것은 집적회로내에 저장하는데 큰 기억용량이 요구되는 문제가 제기된다. 둘째는 modular 연산에 소요되는 과도한 계산시간이다. 곱하여지는 수의 자리 수가 증가함에 따라 연속적인 덧셈으로서 이루어지는 곱셈시간이 급격하게 증가하는 것이다.

이 지수계산의 기억용량과 계산시간 부담을 줄이기 위하여 도입한 것이 지수계산 도중에 생성되는 중간곱에 대한 modular 연산이다. 중간 단계에서 발생하는 숫자에 대하여 어떤 숫자의 modular 값 즉 나누기의 나머지만을 취하여 줄여감으로써 첫째, 요구되는 기억용량의 크기를 일정하게 유지하고, 둘째, 비교적 작은 숫자에 대한 modular 연산이 큰 숫자에 대한 연산보다 빠르므로 결과적으로 계산 속도를 줄이는 것이다. 여기에 더하여 redundant number 체제를 도입하여 덧셈과정의 시간소모 요인이 되는 carry 전파를 제한하고, 이를 통하여 계산속도를 높이는 것이다. 이제 이 modular 곱셈을 좀 더 자세히 서술하고자 한다.

Redundant number system을 채택하게 되면, 덧셈이나 뺄셈과정에서 발생하는 carry나 borrow bit의 전파를 효과적으로 제한시킬 수 있고, 그 결과 덧셈과 뺄셈에 소용되는 시간은 획기적으로 단축시킬 수 있게 된다. 즉 숫자표기에 매 자리에 sum bit(s_i)와 carry bit(c_i)를 표시할 수 있도록 2 bit 저장능력을 부여하는 것이 일반적인 방식인데, carry가 addition 후에 해당하는 자리에 저장되므로 carry의 전파가 없게 되고 따라서 덧셈동작을 minimum gate delay내에 해결할 수가 있다. 즉 어떤 숫자 B를 더하는 과정은 모든 i에 대하여 다음과 같다.

$$s_{i+1}=s_i \oplus c_i \oplus b_i$$

$$c_{i+1}=s_ic_i \vee s_ib_i \vee c_ib_i$$

이 redundant number format은 연속적인 덧셈이 수행되는 경우 carry 전파가 요구되기 때문에, 이를 재조정한 것이 s_i, c_i 대신에 다음과 같이 t_i와 d_i를 쓰는 방법이다[7].

$$t_i=s_i \oplus c_i$$

$$d_{i+1}=s_ic_i$$

이 표현에 의한 숫자와 일반적인 binary number B의 덧셈은 다음과 같이 수행되는데, 여기서 $d_{i+1} \cdot t_i$은 모든 i에 대하여 항상 0이고 $d_0=0$이 된다.

$$s_i=t_i \oplus d_i \oplus b_i$$

$$c_{i+1}=t_ic_i \vee t_ib_i \vee b_id_i$$

위와 같은 사실은 이 redundant number 표기체제 하에서는 연속적인 덧셈에 의해서 carry의 전파가 인접 bit에만으로 국한된다는 사실이다. 즉 어떤 binary number A의 매 자리의 표기를 a_i와 α_i의 pair로 나타내는 경우, 이 a_i를 위의 t_i에 α_i를 d_i에 대응시켜 놓으면 A는 $\sum_{i=0}^{n-1}(a_i \oplus d_i) \cdot 2^i$이 될 것이고 이를 이용한 덧셈계산은 다음과 같이 수행할 수 있게 된다.

먼저 A와 B가 각각 길이 n인 redundant number format의 register이며 a_i와 α_i쌍과 b_i와 β_i쌍이 각각 A와 B의 i번째 자리의 두 bit라고 하자. 그러면 A와 B에 저장된 두 수의 곱 D는 다음과 같이 계산된다.

$$D \leftarrow D+a_{n-1} \cdot 2^{n-1}B$$

$$D \leftarrow D+a_{n-2} \cdot 2^{n-2}B+\alpha_{n-1} \cdot 2^{n-1}B$$

$$\vdots$$

$$D \leftarrow D+a_1 \cdot 2B+\alpha_2 \cdot 2^2B$$

$$D \leftarrow a_0 \cdot B+\alpha_1 \cdot 2B$$

여기서 이 redundant number의 곱셈이 기존의 carry save addition 보다 유리한 점이 발견된다. 그것은 항상 $a_i\alpha_{i+1}=0$이기 때문에, a_i2^iB 또는 $\alpha_{i+1}2^{i+1}B$ 중의 하나가 0이 되어서 계산이 간단해진다는 것이다. 즉, 일반적인 carry save 덧셈의 경우 $a_i\alpha_{i+1}=1$이 될 수 있지만, 이 redundant number system의 경우 $a_i\alpha_{i+1}=0$이 되기 때문에, 각 단계에서 B의 shift와 한개의 addition만 하면 된다는 것이다. 이것이 곱셈 과정에서의 속도개선을 달성할 수 있게 해주는 근거가 된다.

이제 modular 연산을 수행하여, $E\equiv D \bmod C$가 얻어지도록 한다. 여기서 C는 modular값을 정하는 변수이다.

If$(D\geq 2^{n-1}C)$ then $D \leftarrow D-2^{n-1}C$
If$(D\geq 2^{n-2}C)$ then $D \leftarrow D-2^{n-2}C$
.
.
.
If$(D\geq 2C)$ then $D \leftarrow D-2C$
$E \leftarrow D$

이 과정은 전형적인 나눗셈 algorithm이다. D를 dividend로 잡고, C의 2^{n-1}승한값을 divisor로 잡아서 계속적으로 감산해가는 방법이다. 여기서 D가 아주 크게 되면 곱셈의 경우에서처럼 긴 단계를 거쳐야 한다. 이 단계를 줄이는 것이 modular 곱셈의 속도 개선에 큰 도움이 된다.

이 곱셈방법을 쓰면 fast exponentiation을 수행할 수 있다. 즉 iterative addition을 수행하는 과정에서 carry propagate가 억제되어 있기 때문에 operand의 크기에 상관없이 4 gate delay내에 매 iteration을 완결시킬 수 있게 된다. 따라서 곱셈 자체의 256자리의 경우 256 shift 과정을 통하여 수행된다. Exponentiation은 2승, 4승, …, 2^n승의 값을 발생시키고 이들을 다시 곱하므로써 만들어지게 되는 것이다. 이것들은 다음장에서 설명하는 multiple input linear feedback shift register를 이용하여 쉽게 만들 수 있다.

이제 modular 곱셈과정을 설명하고자 한다. 우선 이 연구에서는 입력되는 bit에 대하여 4 bit 단위의 bit군으로 연산을 수행하였는데, 이것은 회로설계상의 최적화 단위를 설정하여 chip 제조시 silicon 면적 최소화와 speed 향상을 기하는 방법이기 때문이다. 일반적으로 전체 bit 단위를 g개의 부분으로 나누어 그 각각이 k bit로 구성되었다고 할 때, 곱셈 operand는 gk bit로 구성된다. 즉 $D_i=\sum_{j=1}^{g}D_{i,j}\cdot 2^{(J-1)}$ 가 된다. 여기에 register R을 결과저장을 위한 register로 사용한다면 전반적인 modular 알고리즘은 그림 2. 1과 같이 표현된다.

이 곱셈결과에 대한 modular operation을 수행하기 위하여도 C의 값과 4 bit 단위의 비교를 수행해야 하는데 전체 bit 비교를 위하여는 다음절에서는 설명하는 비교기의 설계가 전체 암호 계산시간의 결정에 큰 영향을 미치게 된다. Modular 곱셈의 속도를 개선하고 중간곱의 저장에 필요한 기억용량을 줄이기 위하여, Brickell등이 사용한 algorithm[7]의 오류가 정정된 것이다[5].

나. 비교회로의 설계

고속 암호체제 구현에 필요한 알고리즘 개선 및 검증을 하였다. 알고리즘 개선은 modular 곱셈과정에서 발생하는 부분곱의 크기 축소를 위하여 두개의 operand의 크기에 대하여 행하게 되는 비교과정에 적용되었다. 기존의 방법은 최상위 부분 bit에 국한되어 비교하고, 그 이하는 단순감산 또는 가산을 하여 암호화의 정확성에 문제를 갖고 있었다. 이에 대한 개선책으로 상위에서 하위 bit로의 순차비교 방안을 채택하였다. 결국 비교회로가 모든 bit에 대한 처리를 담당하여야 하는데 착안하여, 암호화 속도개선을 4 bit 단위의 계층적 병렬비교를 도입하였다.

이 비교회로는 4 bit 단위의 modular comparator 회로로써 구성하였다. 상위 bit 묶음들의 비교결과가 같을 때만 하위 bit 묶음에 대한 비교가 전체 algorithm에 반영되도록 하여 계산속도를 줄였다. 이 비교기에는 두개의 4 bit operand A, B가 입력되어 대·소 또는 등등여부를 결정하게 된다. 그림 2. 2에는 이 비교기의 회로도가 제시되어 있다.

이 comparator 회로의 많은 bit 비교를 위한 연결은 serial 연결과 parallel 연결이 있다. Serial 연결인 경우 비교결과의 rippling에 의한 delay가 크기 때문에 parallel 연결이 바람직하다. 이 parallel 연결은 bit 수의 증가에 따라 hierarchy의 level이 늘어나게 되

```
Program FAST MULTIPLICATION ;
  Do i=0 to 10                          /* Delayed carry register
     D ← D+A[i]×B ;                        D의 초기화                    */
  End.
  t1 ← 0, t2 ← 0 ;                      /* Control bit값의 초기화        */
  Do j=1 to n                           /* Multiplication & modulo
     B* ← a(n-1)B+αn2B,                    연산 */
     K* → t2 2^11 K+t1 2^10 K,
     D ← 2(D+B*+K*),
     A ← 2A,
     j=j+1 ;
     While (D=2^11 K) do                /* Partial dividend와
          D의 상위부터 4bit씩+2^11 K ;     modular의 크기 비교           */
     If 2^11 K에 대하여 overflow         /* 비교 결과에 따른
          then t2 ← 1,                     control bit t2 설정           */
          else t2 ← 0 ;
     While (D=2^10 K) do
          D의 상위부터 4bit씩+2^10 K ;
     If 2^10 K에 대하여
          overflow이고 t2=0             /* Control bit t1 설정           */
          then t1 → 1,
          else t1 → 0 ;
  End.
Program End.
```

그림 2. 1. 고속 modular 알고리즘에 대한 가상 program

는데 그 level의 수는 총 bit 수 n에 대하여 $\log_4 n$이 된다. 즉 256bit의 경우 매 단계마다 2 gate delay씩 serial 연결이 514 gate delay를 감수해야 하는데 반해, 계층적 병렬비교의 경우 18 gate delay만을 주면 되므로 속도개선 효과가 아주 뛰어난 것을 알 수 있다.

이 회로에 대한 검증은 상업용 논리 simulator인 SILOS상에서 수행하여 성공적인 동작이 확인되었다. 실제 동작 speed 계산을 위하여 transistor 수준의 상세설계가 진행되었고, target 제조공정에 대한 공정변수 정립이 진행되었다. 현재 1.0μ 이하의 최소 선폭을 갖는 고급 CMOS 공정이 요구되는 것으로 판단하고 있다.

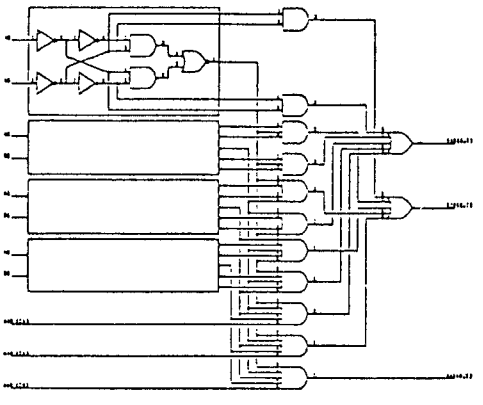

그림 2. 2. 4bit magnitude comparator

3. 암호회로 설계 및 simulation

이 장에서는 공개열쇠 암호계산의 핵심이 되는 fast exponentiation을 수행하는 부분의 회로를 설명하고, 그 회로의 simulation을 수행한 환경 및 결과를 설명하고 있다. Fast exponentiation을 수행하는 회로는 feedback shift register로 구성되었다. 이외에도 주요 회로로써 register file과 control logic이 있다. Register file은 일반적인 microprocessor의 register file로서 그 설계방식이 이미 잘 알려져 있다. 또한 control logic 역시 동작속도에 제한받지 않기 때문에 control flow가 확립되면 그 설계 과정을 자동화 tool에 의하여 처리할 수 있다.

가. Shift Register에 의한 연산

Shift register에서의 logical circuit의 신호는 {0, 1}을 사용하며, 0과 1의 단지 2개의 permutation으로 연산이 이루어진다. 이 연산은 0과 1의 상호변환인데 기존의 값에 1을 더하므로써(modulo 2) 수행된다. 다른 연산은 두 element에 0을 더하므로써(modulo 2) 수행되며, 이때는 상태가 변하지 않는다. 이 shift register의 permutation을 이용하는 cryptosystem이 그림 3. 1에 보여졌다. Sequence $\{s_i\}_{i\geq 0}$이 random하여야 하지만 finite state machine에서 random sequence를 만드는 것이 어렵고, 결국 주기적인 sequence를 형성한다. 따라서, 주기가 길고 올바른 cryptographic 특성을 가지는 random한 sequence를 만들기 위한 시도가 있었다.

Feedback shift register는 binary sequence에 대한 매우 빠른 generator이다. 길이가 n인 feedback register r은 n개의 memory cell로 구성된다. Shift register의 state$(s_0, s_1, \cdots, s_{n-1})$을 형성한다. f가 boolean function이기 때문에 logical switch로 쉽게 구성할 수 있다. 첫번째 shift register는 s_0를 출력할 것이고, state는 $(s_1, s_2, \cdots, s_n)$이 된다. 이때 $s_n = f(s_0, s_1, \cdots, s_{n-1})$이다. 이런 방법으로 계속하면 shift register는 무한(infinite) sequence $\{s_i\}_{i\geq 0}$를 형성할 것이다. 먼저 f가 linear function인 경우를 보자. 즉

$$f(s_0, s_1, \cdots, s_{n-1}) = c_0s_0 + c_1s_1 + \cdots + c_{n-1}s_{n-1}$$

여기서 $C_i \in \{0, 1\}$, $0 \leq i \leq n-1$이고 모든 덧셈은 modulo 2가 된다. Output sequence $\{s_i\}_{i\geq 0}$은 초기값 s_i, $0 \leq i \leq n-1$, 그리고 recurrence relation에 의해 설명될 수 있다. 정의에 의해 $c_n = 1$이며, $s^{(i)}$를 time i에서 state를 표시한다고 하자. 즉, $s^{(i)} = (s_i, s_{i+1}, \cdots, s_{i+n-1})$이면, 다음과 같이 표현된다.

$$s^{(i)} = \sum_{i=0}^{n-1} c_i s^{(i+k)}, \quad k \geq 0$$

Linear feedback shift register(LFSR)의 일반적인 형태가 그림 3. 2에 있다. 여기서 계속 c_i는 feedback coefficient라 한다. 만일 $c_i = 0$이라면 switch는 close된다. c_0는 항상 1이라고 가정한다. 그렇지 않으면 output sequence $\{s_i\}_{i\geq 0}$은 $c_0 = 1$인 LFSR에 의해 형성된 sequence의 delayed version과 같기 때문이다. LFSR의 어떤 결과 state는 유일한 후속 state를 가질뿐만 아니라 유일한 이전 state를 가진다.

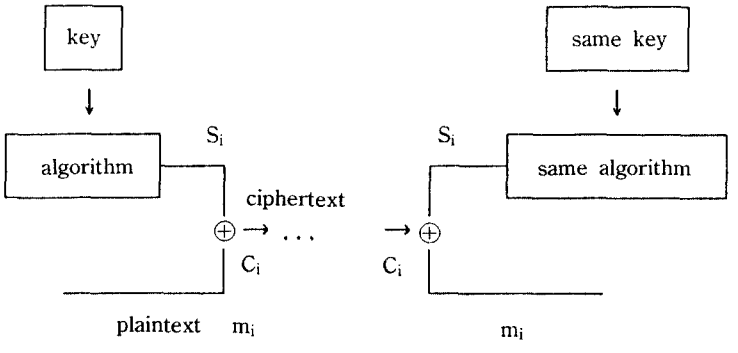

그림 3. 1. Binary cryptosystem with pseudo-random $(s_i)i\geq 0$ sequence

이 LFSR은 여러개의 입력을 갖는 연산에는 시간적인 제약을 가져온다. Fast exponentiation을 위하여는 큰 자릿수의 숫자에 대하여 병렬처리가 가능한 회로가 요구된다. 여기에 사용되는 것이 multiple input shift register(MISR)로서 그 구성은 다음과 같다. 기존의 shift register와 같이 serial 입출력이 있으며, 매 shift register 단계마다 병렬 출력을 갖고 있다. 여기에 각 stage에 연산회로가 첨가되는 것이다.

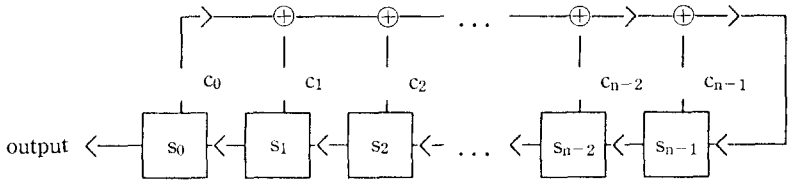

그림 3. 2. General linear feedback shift register

Multiple-input shift register(MISR)를 사용하는 parallel modular operation은 linear feedback shift register 보다 더욱 개선된 속도 특성을 보인다. 이것은 한꺼번에 operand bit가 입력되기 때문이다. 그림 3. 3에서 보여진 MISR 회로는 입출력 bit수만큼의 stage를 갖는다. MISR과 stage 사이에 놓인 modulo 2 adder는 회로의 output 중 하나에 의해 조절된다. Divisor feedback이 또한 적당한 stage에서 adder를 공급한다. 그림 3. 3에 있는 MISR은 divisor polynomial x^5+x^3+x+1를 구현한다. 만일 input 1에서 4까지 0으로 setting 되면, MISR은 LFSR이 된다.

입력 data가 m-stage MISR에 의해 처리되는 m-output circuit을 생각해보자. 연산 cycle i가 전체 sequence의 중간에 놓인 임의의 cycle이라 하자. Cycle i가 회로에 막 적용되고, data가 안정하고, shifting clock이 아직 MISR에 적용 되지 않았다는 의미로서 "연산직후 순간 i"를 정의한다. $R_i(x)$가 cycle의 연산직후 순간 i에서 회로의 m개 data의 sequence를 표현하는 (m−1)차 다항식이라 하자. 그러면

$$R_i(x)=r_{i,m-1}x^{m-1}+r_{i,m-2}x^{m-2}+\cdots+r_{i,1}x+r_{i,0}$$

이다. 또한, $S_{i(x)}$가 연산직후 순간 i에서 MISR의 m개 register의 state를 정의하는 다항식이라 하자. 그러면

$$S_i(x)=S_{i,m-1}x^{m-1}+S_{i,m-2}x^{m-2}+\cdots+S_{i,1}x+S_{i,0}$$

이 된다.

MISR과 연산후 순간 i의 output의 status가 그림 3. 4에서 보여진다. MISR의 state i+1은 다음과 같이 표현된다.

$$S_{i+1}(X)=[R_i(x)+xS_i(X)] \bmod G(x)$$

예를들어, $G(x)=x^5+x^2+x+1$라 하자. 만일

$S_i(x)=x^4+x^2+x+1$, $S_i=10111$
$R_i(x)=x^2+x$ $R_i=00110$

라면 $S_{i+1}(x)=x^3+x^2+x+1$ $S_{i+1}=01111$
이 된다. 이 결과는 다음의 계산에 의해서 보여질 수 있다.

$$\begin{aligned}&[R_i(x)+xS_i(x)] \bmod G(x)\\&=[x^2+x+x(x^4+x^2+x+1)] \bmod G(x)\\&=(x^5+x^3) \bmod G(x)\\&=x^3+x^2+x+1\end{aligned}$$

또는 이 변이를 구현한 그림 3. 5에서 보여지듯이 MISR의 현재 state와 다음 state를 관찰하므로써 위의 결과를 알 수 있다.

MISR의 초기 state S_0는 알려져야 한다. 만일 초기 state가 0이라면, MISR state의 sequence는 다음과 같을 것이다.

$$\begin{aligned}S_1(x)&=[R_0(x)+xS_0(x)] \bmod G(x)\\&=R_0(x) \bmod G(x)\\&=R_0(x)\\S_2(x)&=[R_1(x)+xS_1(x)] \bmod G(x)\\&=[R_1(x)+xS_0(x)] \bmod G(x)\\S_3(x)&=[R_2(x)+xS_2(x)] \bmod G(x)\\&=[R_2(x)+xR_1(x)+x^2R_0(x)] \bmod G(x)\\S_n(x)&=[x^{n-1}R_0(x)+x^{n-2}R_1(x)+\cdots\\&\quad+xR_{n-2}(x)+R_{n-1}(x)] \bmod G(x)\end{aligned}$$

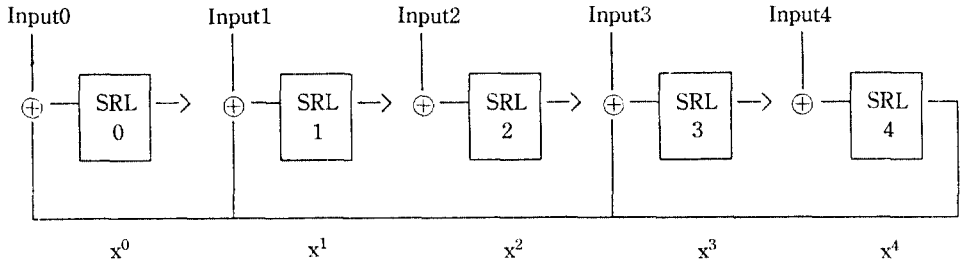

그림 3. 3. Multiple-input signature register(MISR)

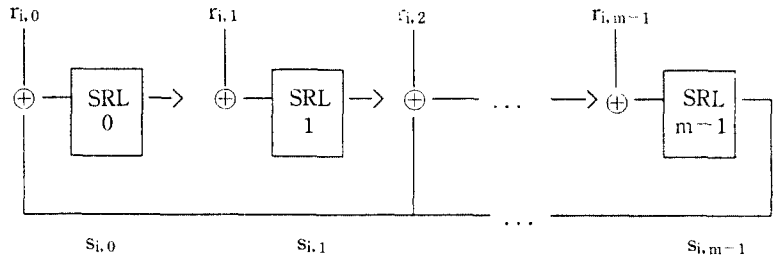

그림 3. 4. MISR after cycle i

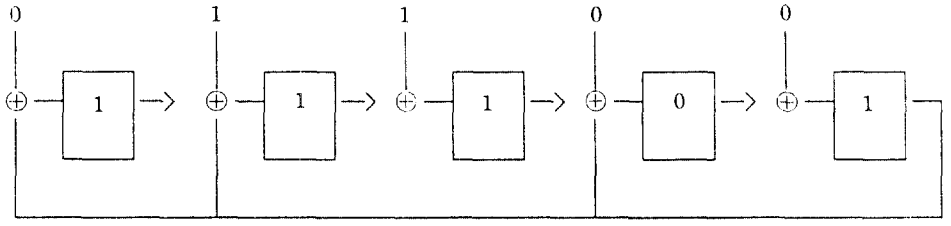

그림 3. 5. MISR at cycle i : $G(x)=x^5+x^2+x+1$,
$R_i(x)=x^2+x$, $S_i(x)=x^4+x^2+x+1$

마지막 식은, n개의 data가 적용된 후 나머지(remainder)이다. 이것은 MISR의 연산이 전체 data bit에 대하여 동시에 적용되기 때문에 LFSR에 비하여 속도의 개선을 이룰 수 있다는 것을 의미한다.

이상에서 설명한 LFSR을 사용하여 fast exponentiation을 수행하는 회로를 구성하였다. 다음절에서 LFSR로 구성된 이 회로를 설명하고, 간단한 동작특성을 설명한다.

나. Fast Exponentiation

Fast exponentiation 알고리즘 개선은 modular 곱셈과정에서 발생하는 부분 곱의 크기 축소를 위하여 행하게 되는 비교과정에 적용되었다. 기존의 방법은 최상위 부분 bit에 국한되어 비교하고, 그 이하는 단순감산 또는 가산을 하여 암호화의 정확성에 문제를 갖고 있었다. 이에 대한 개선책으로 상위에서 하위 bit로의 순차비교 방안이 제시되었다. 결국 비교회로가 모든 bit에 대한 처리를 담당하여야 하는데 착안하여, 암호화 속도개선을 4 bit 단위의 계층적 병렬비교를 도입하였다.

이 비교회로는 4 bit 단위의 modular comparator 회로를 구성하였으며, 상위 bit 묶음들의 비교결과가 같을 때만 하위 bit 묶음 비교가 전체 algorithm에

반영되도록 하여 계산속도를 줄였다. 이 comparator 회로의 많은 bit 비교를 위한 연결은 serial 연결과 parallel 연결이 있다. Serial 연결인 경우 비교결과의 rippling에 의한 delay가 크기 때문에 parallel 연결이 바람직하다. 이 parallel 연결은 bit 수의 증가에 따라 계층구조의 level이 늘어나게 되는데 그 level의 수는 총 bit 수 n에 대하여 $\log_4 n$이 된다. 즉 256 bit의 경우 매 단계마다 2 gate delay씩 serial 연결이 514 gate delay를 감수해야 하는데 반해, 계층적 병렬비교의 경우 18 gate delay만을 주면 되므로 속도개선 효과가 아주 뛰어난 것을 알 수 있다.

여기서 사용되는 두가지의 register type, 즉 R register와 S register는 exponent representation과 key storage에 사용되며 그 길이는 각각 256 bit이다. R은 primary exponent register이고, repeat exponentiation에 사용된다. 256 bit exponent(R register)는 593 bit exponent에 mapping된다. Mapping의 목적은 fast exponentiation에 대한 exponent의 hamming weight를 제한하는 것이다. R register는 repeated exponentiation을 수행하는 256 bit maximal length linear feedback shift register로서 연결된다.

Fast exponentiation을 수행하는 부분의 회로도는 다음 그림 3. 6과 같다. 한스텝의 LFSR에 clock이 low, high로 들어오면 데이타가 들어와서 다음 clock이 다시 low, high로 들어와 다음 스텝의 LFSR로 데이타가 shift되기전까지 저장된다. Clock이 low인 동안 LFSR에 데이타가 들어오고, high가 되면 다음 스텝으로 데이타가 넘겨지기전까지 한 스텝의 LFSR내에서 데이타가 전달된다.

다. 설계 Simulation

Simulation을 통하여 주어진 회로의 전기적 동작 특성을 설계의 물리적 구현이전에 효과적으로 해석할 수 있다. Simulation에는 logic simulation과 circuit simulation이라는 두가지의 형태가 있다. 먼저 logic simulation은 대규모 논리회로의 시간해석을 빠른시간에 처리할 수 있으나, 해석할 수 있는 회로의 형태가 논리회로로 국한되어 있고 해석결과는 논리신호로 표시되며 지연시간과 같은 전기적 변수들의 계산이 생략되어 있다.

Circuit simulation은 transistor 수준의 기능을 점검하는 것으로서 logic simulation에 비해 속도가 100배 정도 느린 것이 보통이다. 그러나 DC analysis, transient analysis, frequency analysis, noise analysis 등이 정확하게 수행되며, 그 오차도 5% 내외이다. 이번에 수행된 회로의 simulation은 clock의 속도에 따른 지연시간이나 전기적 동작 특성을 요하므로 circuit simulation을 통하여 technology의 사용가능성을 주로 검토하였다.

먼저 simulation의 입력 data base가 되는 회로의 선택을 하여야 한다. Circuit simulation은 그 계산시간이 길고 대규모 회로에 대한 해석은 불가능하므로 전 회로에 대하여 simulation을 한다는 것은 비효율적이다. 그러므로 전기적인 특성이 같은 전체 회로의 대응 model을 추출하여 characterization model로 사용한다. Simulation을 위한 characterization model은 회로의 critical path 즉 worst case timing path에서의 speed와 operation이 예측되도록 만들어진 회로이다. Fast exponentiation을 수행하는 부분의 두 register S와 R중에서 R register에 대한 simulation을 수행하였다. R register는 LFSR로 구성되었으며, fast exponentiation에서 가장 시간을 많이 소요하는 회로 동작특성을 갖기 때문에 이 register에 대해서만 simulation을 수행하였다. R register의 characterization model이 그림 3. 7에서 보여지고 있으며, 두개의 연속적인 LFSR로 구성되어 있다.

암호 IC의 음성대역내에서 실시간 처리를 위해서는, 동작속도의 평가가 우선되어야 한다. 회로의 동작속도 계산을 위한 simulation은 4가지 반도체 제조공정, 0.8μ, 1.2μ, 1.5μ, 3.0μ의 최소선폭을 갖는 CMOS 공정별로 simulation을 수행하였다. 0.8μ와 1.2μ 공정에 대한 simulation은 1.5μ의 model parameter를 사용하여 channel length만을 바꾸어서 수행되었다. 이것은 channel length의 변화가 회로동작에 가장 심각한 영향을 미치기 때문이다.

Simulation에 있어서 두 clock cycle을 기본 operation으로 잡았는데, 이것은 R register의 buffering 특성이 두 clock cycle을 한개의 data 계산시간으로 잡아야 하기 때문이다. 즉, 한 스텝의 LFSR에 clock이 low, high로 들어오면 데이타가 들어와서 다음 clock이

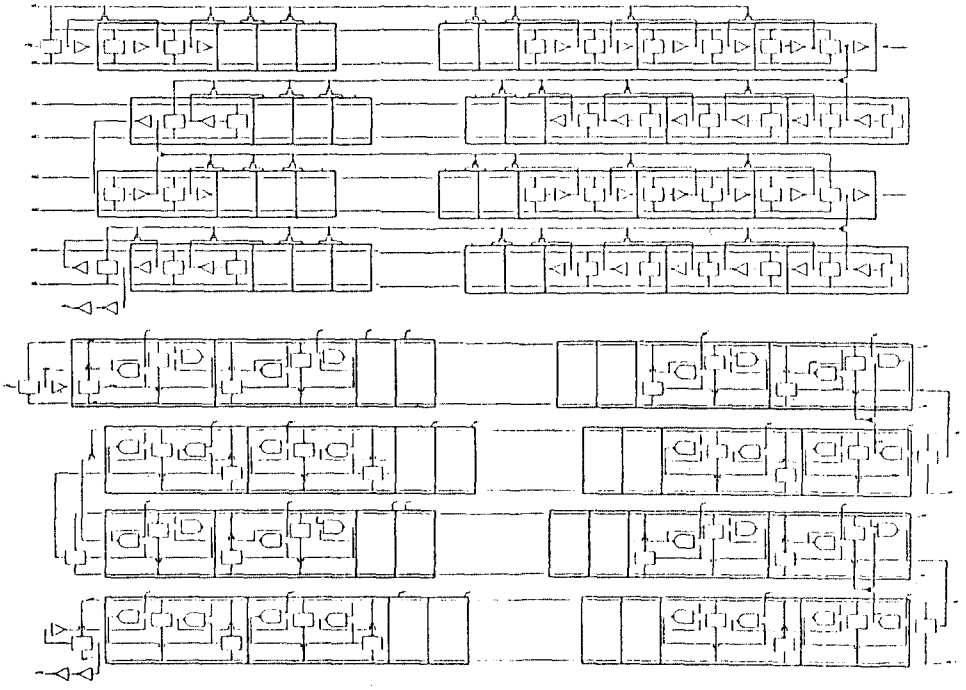

그림 3. 6. Fast exponentiation circuit

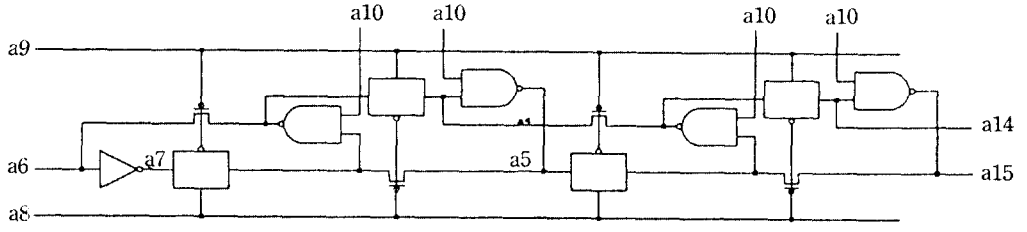

그림 3. 7. Simulation model circuit

다시 low, high로 들어와 다음 스텝의 LFSR에 데이타가 shift되기전까지 저장된다. Clock이 low인 동안 LFSR에 데이타가 들어오고, high가 되면 다음 스텝으로 데이타가 넘겨지기 전까지 한 스텝의 LFSR내에서 데이타가 전달된다. 다음 스텝의 LFSR로 data가 전달되기까지는 두 clock이 필요하다. Shifting operation에 대한 한 clock 주기가 그림 3. 8과 같은 경우 최소한의 data cycle은 single rising time의 8배가 되어야 한다.

이 결과를 음성 서비스에 대한 암호화 기능성을

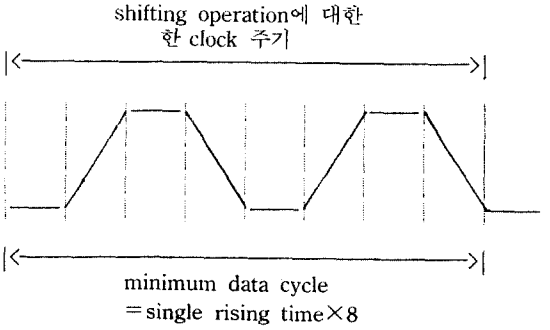

그림 3. 8. Minimum data cycle

보기로 하자. 전화망 음성주파수는 channel당 400 Hz이고 이것을 sampling하면 주파수의 두배이므로 8000 sampling/sec가 된다. 이것을 매 sample마다 8bit로 encoding하면 64kbit/sec가 된다. 이 64kbit /sec의 data rate가 전자교환기 등의 기본 교환단위가 되고 있다. 따라서 암호화는 이 정도 이상의 data 양을 처리할 수 있는 속도를 내야만 음성서비스에 적용 가능하다. 이 속도는 암호화 칩을 제작하는 반도체 기술에 의하여 결정되는데, 이 연구의 핵심은 어느 정도의 수준의 반도체 공정기술과 설계방법이 타당성 여부를 검토하는 것이다.

기존의 암호칩에 대한 자료에 의하면 현재 3μ 정도의 CMOS technology로 약 19.2kbps 정도까지만 처리가 가능하다. 따라서 음성서비스에 필요한 64kbps는 기존의 칩의 속도의 약 3.3배 정도이므로 회로 simulation을 통하여 이 속도를 달성할 수 있는 공정변수를 찾아내야 한다. 이를 위하여 우리는 4가지의 공정변수셑을 대상으로 가장 시간특성에 제약이 큰 회로에 대하여 회로 simulation을 실시하였다.

이 simulation에는 가장 널리 통용되는 SPICE가 사용되었으며, 공정변수 중에서도 특히 속도에 가장 큰 영향을 미치는 gate의 channel length를 각각 3.0, 1.5, 1.2, 0.8 micron으로 변화시켜가며 실시하였다. 이 각각의 최소 선폭은 기존의 암호칩이 주로 3.0과 1.5micron 공정을 사용하였기 때문에 여기서 simulation 결과와 문헌상에 기록된 처리속도를 correlate하기 위한 것이며, 1.2micron과 0.8 micron은 각각 256 k DRAM과 4M bit DRAM의 기술로서 현재 국내 논리칩 제조에 사용되는 최신 공정인 것이다. 이 연구는 이들 최신 공정기술의 적용의 최하한선도 아울러 정의하고자 하는 것이다.

여기서 속도계산에 사용하는 최소의 시간단위는 신호파형의 rise time이나 fall time이 기준이 된다. 이 두 time 중에서 긴 시간을 잡아서 파형의 rising, leveling, falling에 요구되는 시간간격을 같은 시간으로 잡아서 계산하게 된다. 그림 3. 8은 이 세 time 간격을 보이고 있다. 여기서 simulation의 대상이 되는 shift register stage는 8개의 time 간격을 요구한다. Simulation 출력에서 관측되는 rising time이나 falling time 중에서 긴 시간의 8배를 해서 기준시간으로 잡게된다.

Simulation의 수행결과로 clock speed가 0.8μ의 경우 2000MHz, 1.2μ의 경우 1250MHz, 1.5μ의 경우 600MHz, 3μ의 경우 100MHz에서 회로동작에 한계가 있음을 알 수 있었다. 이상 네가지 공정의 clock speed에 대하여 비교한 도표가 그림 3. 9와 같이 나타난다. 이 결과가 제시하는 것은 음성주파의 처리를 위해서는 최소한 1.0 micron 이하의 최소선폭을 갖는 반도체 설계기법과 공정이 요구된다는 것이다.

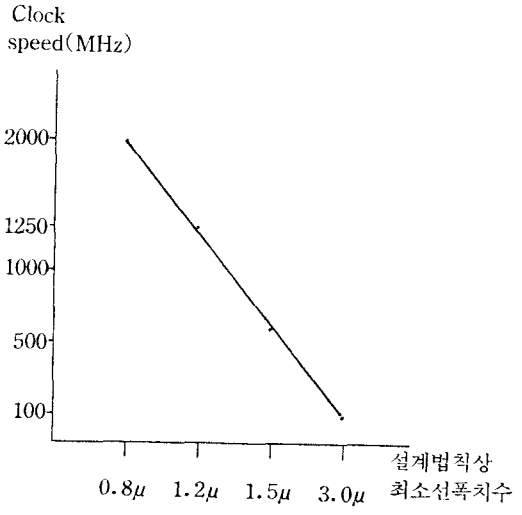

그림 3. 9. 공정에 따른 Clock speed 비교

라. 기존 암호칩 분석

분석의 대상이 되는 기존의 암호회로의 구조는 data path를 이루는 exponentiation logic과 register file 그리고 control path로 구성되어 있다[8]. 전체 규모는 6만 gate 이상의 논리회로 부담을 갖고 있는데, 이것은 기존의 RSA방식 chip의 분석을 토대로 계산한 것이다. 이 연구에서 목표로 하고 있는 음성분야의 service를 위하여 필요한 경우 여러개의 exponentiation logic이 사용되고 이를 pipeline화 할 수도 있게 된다. 이 경우에는 매 pipeline 추가에 약 8천 gate가 더 소요된다. 따라서 4 stage pipeline의 경우 전체 경우는 10만 gate 이상을 요구하는 것이다.

이런 경우의 집적회로 설계에서는 몇 단계의 block별 prototyping을 통하여 기능적 완벽성과 처리속도, 전기적인 특성파악이 우선되어야 한다. 기존의 IC들에 대한 reverse engineering을 실시하는 목적은 이 과정을 통하여 목표 device의 처리속도 반도체 공정상의 요구사항 정리 그리고 효과적인 회로설계 방식을 터득하고자 하였다. 이것은 정확한 simulation의 기본자료가 되며, prototyping을 거치지 않고도 성공적인 기능 구현을 가능케 하는 설계기법인 것이다.

Reverse engineering은 chip사진에 나타난 공정후의 layout pattern으로부터 먼저 transistor level의 회로를 추출하고, 이를 토대로 logic level 그리고 block level의 구조를 찾아내는 과정이다. 즉, 물리적인 구조체에서 기능적인 구조를 찾아내는 bottom-up 방식의 분석인 것이다. 기존의 chip에 대한 reverse engineering을 위하여 package를 제거한 bare chip에 대한 현미경 사진의 확보, chip의 상층부의 연결을 위한 metal과 poly의 적층구조를 차례로 제거시킨 die 등을 준비하였다. 이 과정은 미국내의 전문업체가 담당하여 상부 metal층 제거, passivation층 제거를 수행하였고, 이렇게 각층이 제거된 chip의 표면에 대하여 전체적으로 3,600매의 die 사진촬영을 하였다. 이 사진들은 mosaic 형태로 400배 확대된 전체 chip의 구조를 제공하고 있는데, 이들 중에서도 특히 shift register를 중심으로 하는 지수연산 logic을 중점적으로 분석하여, 이론적인 회로 model과 chip상에 구현된 실제를 비교, 분석하였다.

Chip사진으로 부터의 회로추출은 다음과 같이 수행된다. 우선 가장 분별하기 쉬운 power line을 먼저 찾아낸다. Vdd line과 ground line의 배열은 각 block별 경계를 규정짓는 효과적인 방법이다. 대형 chip의 경우 여러사람의 공동작업의 결과인데 power line의 공유는 지극히 어려워서 자연스럽게 block간의 경계선을 이루게 된다. 두번째 단계는 block간의 신호선을 규정짓게 되고 이들 신호선의 의미가 확실해지면 내부 logic의 내역은 어느 정도 윤곽이 드러나게 된다

규칙적인 array 구조의 설계분석 작업은 cell의 분석과 이의 반복적인 연결을 통하여 손쉽게 이루어진다. 최종단계는 random logic으로써 gate별 추출과 연결회로에 대한 분석이 이루어지게 된다. 먼저 규칙적인 array의 cell에 대한 분석은 우선 cell boundary에서의 신호선 정의와 transistor의 구조를 찾아내는 것이다. 신호선은 모든 cell에 공통으로 연결되어 있는 신호선, 예컨대 clock, reset 등을 규정한 후 cell간의 data 연결선을 정의하게 된다. 또한 transistor를 찾아내는 것은 chip 사진상에 나타난 diffusion 또는 ion implantation 지역 위에 걸쳐 있는 polysilicon을 gate로 잡고, power line에 가까운 diffusion 지역을 drain으로 잡고, ground에 가까운 지역을 source로 잡는 과정을 거친다. 여기에 source와 drain 위에 metal 층과의 contact 유무에 따라 source와 drain의 연결을 찾아내게 된다.

이 transistor간의 연결내역 추출과정은 우선 chip상의 각종 feature의 기하학적 topology에 따라 회로도를 작성하게 되고, 이를 토대로 입출력 신호를 정리한 transistor level의 회로도를 완성하는 것이다. 물론 이 과정에서 까다로운 공정이 개입되는 경우 device의 단면도가 요구되기도 한다. 대부분의 대형 chip은 random logic 설계에 있어서 channelled gate array이나 standard cell 설계방식을 활용하게 되므로 각 signal net에 대한 naming과 transistor의 연결선을 체계적으로 추적하여 전체배선을 완성해가는 것이다.

Transistor level의 회로추출이 끝나면, 이 결과를 logi level 회로도 작성의 기본자료로 활용한다. 이 과정은 CMOS logic의 경우 n-transistor logic의 gate level logic화를 진행시키는데, transistor의 series/parallel 연결망을 logic으로 전환한 후에 feedback path를 찾아서 sequential logic으로 통합하는 것이다. 여기서 n-transistor logic은 standard CMOS의 경우 p-transistor logic과 논리적인 dual 관계에 있기 때문에, p-transistor logic의 추출결과는 n-transistor logic의 정확성을 검증하는 자료로서 활용된다. 또한 대부분의 sequential 회로요소는 회로동작속도를 유지시키기 위하여 기하학적으로 근접시켜 설계하기 때문에 이것을 transistor level 회로도에서 추출하는데 문제가 없다.

이 각 단계는 simulation을 통한 검증과정을 거치게 되는데, 이 과정에서 block간의 신호교환 sequence를 정확히 알게 되면 완벽한 회로 도면을 확보하게 해주는

것이다. 이 simulation은 두가지가 있는데, 첫째는 circuit simulation으로서 SPICE를 이용하는 logic timing과 기능의 정상적인 동작 여부가 동시에 검증된다. 특히 이 암호칩의 입출력 회로와 고전압 회로, 저전압 회로, dynamic logic 그리고 speed-critical circuit의 추출결과 확인에 이 simulation 과정은 없어서는 안되는 과정이다. 둘째는 logic simulation으로서 주로 논리기능의 integrety를 검증하게 된다. 대부분의 block 별 circuit 추출의 정확성은 logic simulation을 통하여 확인된다.

그림 3. 10은 shift register stage의 cell pattern을 보여주고 있다. 여기에는 이 cell 내부에 있는 모든 transistor의 위치와 연결관계가 포함되어 있고, 이것을 정리하면 그림 3. 11과 같은 transistor 회로가 얻어지게 된다. 이 transister level 회로는 다시 logic 회로의 추출대상이 되는데, 그림 3. 12에는 이 cell의 gata level의 회로도를 보이고 있다. 이 cell이 256개 연속으로 연결되었으며, 이 cell들간의 신호교환을 확인하여 하나의 커다란 function block이 완성된다. 공정의 mask sequence는 그림 3. 13에 나타내었다. 이 chip은 double metal single poly의 배선층을 갖고 있으며, 1.5 micron의 최소선폭을 사용하는 설계법칙을 가지고 제작되었다.

이렇게 해서 얻어진 회로가 그림 3. 14이다. 이것은 우리가 목표하고 있는 multiple input shift register와 유사하나, 속도 개선을 위하여 제한된 숫자의 polynomial만을 사용할 것을 전제로 과감한 회로생략이 이루어져 있는 것이 발견되었다. 이로써 MISR을 기초로 한 fast exponentiation이 실제 적용과정에 무리가 없는 것이 판명되었으며, 특히 0.8μ의 회로선폭을 갖는 CMOS 공정으로 RSA 방식의 암호 chip을 구현할 수 있다는 결론에 도달하게 되었다.

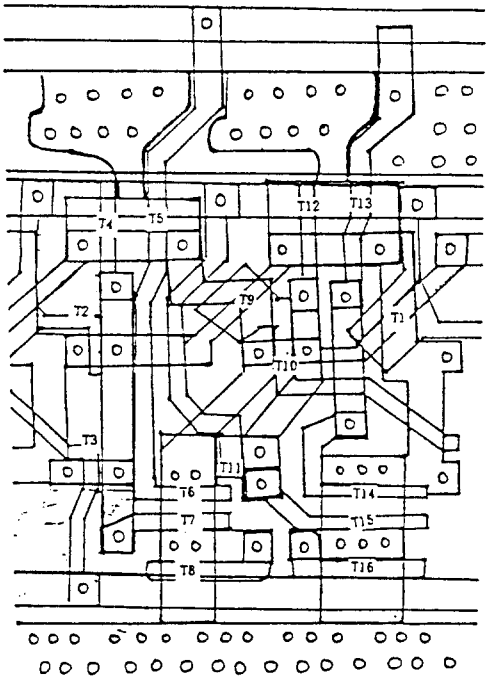

그림 3. 10. chip 사진상의 pattern과 transistor의 위치

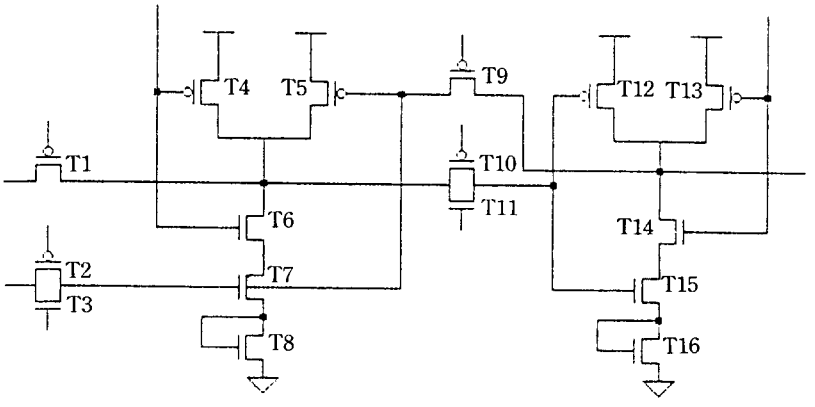

그림 3. 11. chip 사진의 transistor 연결회로도

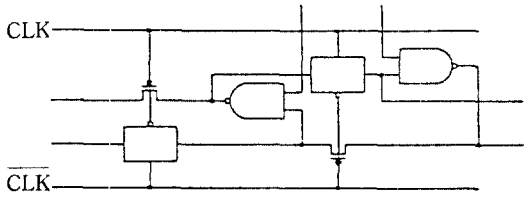

그림 3. 12. Gate level cell

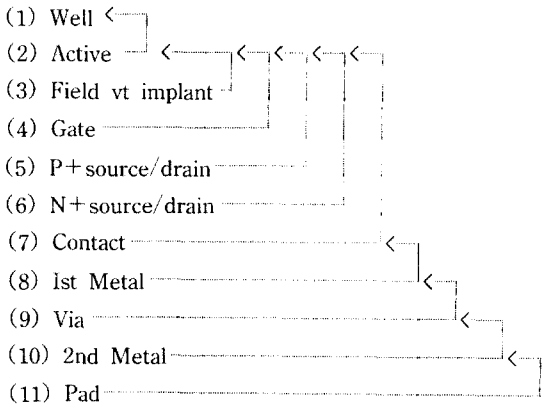

그림 3. 13. Mask sequence of a double metal single poly cryptochip

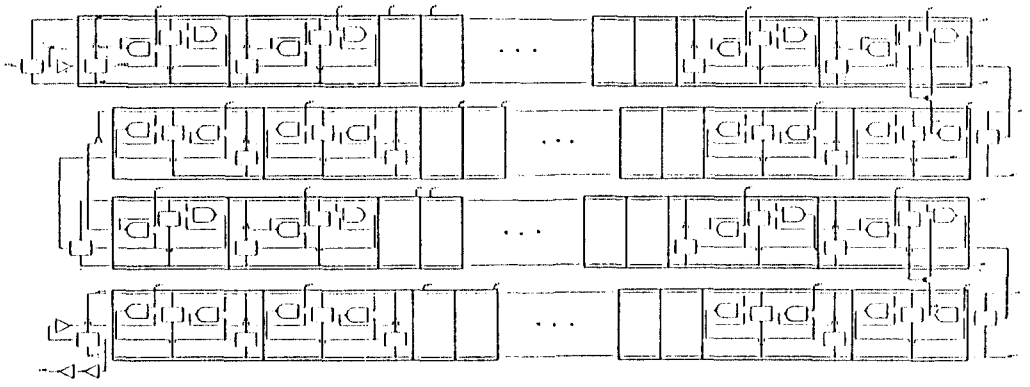

그림 3. 14. Linear feedback shift register

4. 결 론

정보통신체제의 발전은 음성전화망의 확충과 함께 정보의 생산과 유통에 장족의 발전을 가져오고 있다. 공중통신망을 이용하는 정보의 유통과정에 있어서 정보가치의 유지는 정보의 노출방지, 정확한 송수신 시간의 보안체제확립의 바탕위에 가능하다는 인식하에 정보보호의 방식이 도입되고 있다. 따라서 정보보호는 정보통신체제 발전의 핵심으로서 서비스의 품질을 높이고 정보의 유통을 산업화하는 주요 수단이 되고 있다.

정보 보호의 수단으로서 암호는 중요한 역할을 담당하는데, 이것은 전기통신망 운영체제에 적은 비용으로 적용할 수 있으며, 막대한 경제적 손실을 유발시킬 수 있는 정보 유출을 방지키켜 주게 된다. 특히 반도체 집적회로 설계 및 제조기술의 발전과 컴퓨터 기술의 발달은 이 암호기술이 특정용도에 국한하지 않고 일반 대중에 값싸게 제공될 수 있는 길을 터 놓았다. 이 결과 과거에는 계산수단의 비용의 벽에 부딪혀 사용할 수 없었던 고급 암호 기법들이 활용될 수 있게 된것이다.

이 연구는 음성정보 서비스를 근간으로 하는 공중 전화망에 암호화 기법을 도입하기 위한 암호 칩 설계에 촛점을 맞추었다. 특히 64Kbps의 data rate를 갖는 전화시스템에서 음성의 실시간 처리를 위하여, 고속 암호칩 설계에 필요한 설계법칙과 공정 사양의 하한선을 설정하는데 주력하였다. 이는 과거의 암호 칩들이 기껏해야 19.2kbps의 비음성 통신에만 적용가능한

처리속도를 가지고 있기 때문에, 그 사용 범위가 지극히 제한되어 있었다.

지수계산의 시간단축을 위하여 네가지의 방안이 추구되었다. 첫째 계산과정의 숫자 표현을 redundant number system을 도입하여 사용하고, 둘째 지수계산과정에서 modular 연산을 병행하므로써 비교적 작은 규모의 숫자에 대한 연산을 하도록 하였으며, 셋째 숫자 표현과정의 각 digit에서 carry의 전파를 억제하는 표기 방법을 채택하고, 네째 고속 반도체 집적회로 특성을 활용하기 위한 최소한의 설계법칙과 공정을 채택하도록 하였다. Redundant number system을 이용하는 알고리즘이 정리되었고, 특히 modular operation시에 비교과정이 기존의 것보다 개선된 algorithm을 채택하였다.

또한 exponentiation logic의 핵심 회로 요소인 shift register stage의 transistor level 회로에 대하여, 네가지 설계법칙을 적용한 circuit simulation을 수행하였다. 여기에 적용된 설계법칙은 3.0μ, 1.5μ, 1.2μ, 그리고 0.8μ으로서, 3.0μ과 1.5μ의 설계법칙은 기존의 chip과 속도 correlation을 위하여 사용하였고 상대적인 scaling에 의하여 1.2μ과 0.8μ의 설계법칙을 검토하였다. 그 결과 음성 service에 대한 공개열쇠 암호화의 구현을 위하여는 1.0μ이 하의 설계법칙을 가져야 하는 것으로 판명되었다. 또한 modular operation에서의 비교연산을 위하여 4 bit 단위의 sequential 비교 회로를 구사하여 speed 개선과 연산결과의 정확성을 유지시켰다. 이 회로 역시 logic simulation에 의하여 그 기능이 검증되었다. 이것은 modular multiplication 과정에서 발생하는 중간결과의 숫자 크기를 줄이기 위한 modular operation에 요구되는 것이다. 이 결과는 기존의 저속 암호 IC의 reverse engineering을 통하여 그 기능의 타당성이 확인되었다.

참 고 문 헌

1. 유영갑등, Fault Tolerant Cryptography용 IC개발, 서울대학교 반도체공동연구소 보고서 ISRC 90-E-DE-C002, 1990.

2. 이선복, 유영갑, "Public key cryptography의 PC communication adapter에 관한 연구", 충북대학교 산업과학기술연구소 논문집, 제 4 권 제 2 호, pp. 101-106, 1990.

3. 임채호, 변옥환, "정보통신 시스템에서의 시큐리티", 월간 전자과학, 제 30 권, 통권 344 호, 134-156쪽, 1988년 1월호.

4. 한선경, 유영갑, "고속 modular 곱셈 알고리즘 연구", 충북대학교 산업과학기술연구소 논문집 제 4 권 제 2 호, pp. 8-92, 1990.

5. 한선경, 이선복, 유영갑, "공개키 암호체계를 위한 modular 곱셈개선과 통신회로 구현에 관한 연구", 한국통신학회지 16권 7호, 651-662쪽, 1991년 7월.

6. R. Akiyama, A. Yamashita and H. Nakamura, "A Study on microprocessor controlled DES crypto equipment using the public key distribution system", IECE of Japan, Vol. CS 81-65, pp. 43-48, July, 1981.

7. E. F. Brickell, "A fast modular multiplication algorithm with application to two key cryptography", Proc. CRYPTO-82, Santa Barbara, pp. 51-60, Aug, 1982.

8. Calmos Semiconductor Inc., "CA34C168 Data encryption processor data sheet", CALMOS Inc., 1988.

9. W. Diffie and M. Heliman, "New direction in cryptography", IEEE Trans. Info. Theory, Vol. IT-22, No. 6, pp. 644-69, Nov, 1976.

10. T. Elgamal, "A public key cryptosystem and a signature scheme based on discrete logarithm", IEEE Trans. Inform. Theory, Vol. IT-21, No. 4, pp. 473-481, July, 1985.

11. H. Gunji and D. arnon, "On polynomial factorization over finite fields", Mathematics of Computation, Vol. 36, No. 153, pp. 281-287, Jan, 1981.

12. K Hirose, "Prime numbers and factorization into prime numbers on the relation of cryptography", 일본 전자통신 학회지, Vol. 69, No. 4, 5, pp. 462-468, May, 1986.

13. Y. Iwadare, "The algorithms in cryptography", 일본전자통신학회지, Vol. 69, No. 5, pp. 462-468, May, 1986.

14. D. E. Kunth, The Art of Computer Programming, Vol. 2 : SemiNumerical Algorithm, Addison-Wesley, Reading, Mass., 1969.

15. M. Kochanski, "Developing an RSA chip", Business Simulations Ltd., Kent, England, 1984.

16. S. Miyaguchi, "Fast encryption algorithm for the RSA cryptographic system", IEEE CH1792-2, pp. 672-678, 1982.

17. M. Morii and M. Kasahara, "New public key of cryptosystem using logarithms over GF(p)", 일본전자통신학회 논문지, Vol. J71-D, No. 2, pp. 448-453, Feb, 1988.

18. C. Muller-Schloer, "A Microprocessor-based cryptoprocessor", IEEE Micro, Vol. 3. No. 5, pp. 5-15, October, 1983.

19. G. A. Orton et al., "VLSI implementation of public key encryption algorithm", Proc. CRYPTO-86, Santa Barbara, pp. 275-301, Aug, 1986.

20. J. C. Pailles and M. Girault. "The security processor C. R. I. P. T.", Proc. FIPS-86, pp. 127-139, 1986.

21. R. L. Rivest, A. Shamir and L. Adleman, "A method for obtaining digital signatures and public key cryptosystem", ACM Comm., Vol. 21, No. 2, pp. 120-126, Feb, 1978.

22. R. L. Rivest, "A description of a single chip implementation of the RSA public key cryptosystem", NTC-1980, IEEE CH 1539-6, 49. 2. 1-49. 2. 5, 1980.

23. C. Ronse, Feedback Shift Registers, Lecture Notes in Cimputer Science, Vol. 169, Springer Verlag, New York, 1982.

24. RSA Security Inc., "Data sheet on RSA chip 6720/5120/3360 COMS/VLSI Data cryptographic processor", RSA Security Inc. Jan, 1985.

25. J. Seberry and J. Pieprzyk, Cryptography : An Introduction to Computer Security, Prentice Hall, New York, 1989.

26. H. C. A. van Tilborg, An Introduction to Cryptology, Kluwer Academic Publishers, Norwell, Massachusetts, 1988.

27. U. S. Dept. of Commerce, National Bureau of Standards, Data Encryption Standard, Federal Information Processing Standard, Pub. 46, 1977.

28. S. Vassiliadis, M. Putrino and E. M. Schwang, "Parallel encrypted array multipliers", IBM J. Res. Develop., Vol. 4, pp. 536-551.

29. T. Yamamoto and R. Akiyama, "A data encryption device incorporating fast PKDS", IEEE 83 CH19567-2, pp. 1085-1090, 1983.

□ 著者紹介

劉 泳 甲

1948년생
1975년 8월 서강대학교 전자공학과(공학사)
1981년 8월 미국 미시간대학교 전기전산학과(공학석사)
1986년 4월 미국 미시간대학교 전기전산학과(공학박사)
1975년 8월～1979년 8월 국방과학연구소 연구원
1982년 4월～1986년 4월 미시간 전산연구소
1986년 2월～1988년 2월 금성반도체(주) 책임연구원
1988년 3월～충북대학교 정보통신공학과 학과장
1992년 4월～충북대학교 정보통신산업연구소장
주관심분야 : 반도체 집적회로테스트, 고장극복형 컴퓨터 구조, 가변익 항공기 제어, 중·대형 컴퓨터 제작 및 제조기술, 정밀인쇄장치 구조설계

□ 著者紹介

韓 善 景

1991년 2월 충북대학교 정보통신공학과(공학사)
현재 충북대학교 대학원 정보통신공학과 석사과정