최대 임계 지연 크기에 따른 SHA-1 파이프라인 구성★

이 제 훈*, 최 규 만**

*강원대학교 공학대학 전자정보통신공학부, **가톨릭관동대학교 공과대학 전자공학과

요 약

본 논문은 SHA-1 암호 알고리즘의 최대 임계 지연과 유사한 연산 지연을 갖는 새로운 고속 SHA-1 파이프라인 구조를 제안한다. 기존 SHA-1 파이프라인 구조들은 하나의 단계연산 혹은 언폴딩된 단계연산에 기반한 파이프라인 구조를 갖는다. 파이프라인 실행에 따른 병렬 처리로 성능은 크게 향상되나, 라운드의 모든 단계연산을 언폴딩하였을 때와 비교하여 최대 임계 지연의 크기가 증가한다. 제안한 파이프라인 스테이지 회로는 라운드의 최대 임계 지연을 반복 연산 수로 나눈 만큼의 지연 시간을 갖도록 구성함으로써, 불필요한 레이턴시 증가를 방지하였다. 실험 결과, 회로크기에 따른 동작속도 비율에서 제안된 SHA-1 파이프라인 구조는 0.99 및 1.62로 기존 구조에 비해 우월함을 증명하였다. 제안된 파이프라인 구조는 반복 연산을 갖는 다양한 암호 및 신호 처리 회로에 적용 가능할 것으로 기대된다.

SHA-1 Pipeline Configuration According to the Maximum Critical Path Delay

Je-Hoon Lee* and Gyu-Man Choi**

*Div. of Electronics, Information and Communication Eng., Kangwon National University
**Dept. of Electrical Engineering, Catholic GwanDong University

ABSTRACT

This paper presents a new high-speed SHA-1 pipeline architecture having a computation delay close to the maximum critical path delay of the original SHA-1. The typical SHA-1 pipelines are based on either a hash operation or unfolded hash operations. Their throughputs are greatly enhanced by the parallel processing in the pipeline, but the maximum critical path delay will be increased in comparison with the unfolding of all hash operations in each round. The pipeline stage logics in the proposed SHA-1 has the latency is similar with the result of dividing the maximum threshold delay of a round by the number of iterations. Experimental results show that the proposed SHA-1 pipeline structure is 0.99 and 1.62 at the operating speed ratio according to circuit size, which is superior to the conventional structure. The proposed pipeline architecture is expected to be applicable to various cryptographic and signal processing circuits with iterative operations.

Key words : Hash algorithm, SHA-1, pipeline, non-linear pipeline, iteration

접수일(2016년 10월 28일), 게재확정일(2016년 12월 17일)
★ 본 연구는 한국연구재단의 기본연구사업으로 수행된 연구결과임(No.NRF-2012H1B8A2026055). 2015년도 강원대학교 대학회계 학술연구조성비로 연구하였음.(관리번호-201510015)

* 강원대학교 전자정보통신공학부 (jehoon.lee@kangwon.ac.kr)
** 가톨릭관동대학교 전자공학과, 교신저자

1. 서 론

최근, 블루투스, 와이파이등 무선 통신 기술의 발달로 데이터 전송 속도 및 전송량이 급증하고 있다. 사물인터넷의 확산, 소셜 네트워크 서비스와 모바일 금융등 개인 정보를 이용한 어플리케이션이 확산됨에 따라 높은 수준의 정보보호가 필수적이다. 해쉬 알고리즘은 단방향 함수로 복호화가 필요없어 고속 동작이 가능하다. 해쉬 알고리즘에 의해 축약된 메시지에 디지털 서명을 결합하여 서명에 필요한 계산량, 메모리 및 전송량을 크게 줄일 수 있고, SHA (secure hash algorithm)-1 해쉬 알고리즘은 메시지 서명, 메시지 인증 및 메시지 무결성등에 주로 사용된다 [1-3].

최근 고속 통신 기술의 발달로 고속 해쉬 연산기가 요구되며, 언폴딩과 파이프라인 구조를 채택한 해쉬 연산 구조들이 제안되었다. SHA-1 해쉬 알고리즘은 임의의 길이의 메시지로부터 80번의 단계연산을 반복 실행하여 160 비트 해쉬값을 생성한다. 언폴딩은 두 개 이상의 단계연산을 하나의 연산으로 묶고, 동시 실행이 가능한 데이터패스를 늘려 성능을 향상시킨다.

파이프라인 구조를 갖는 해쉬 연산기는 전체 데이터패스를 여러 개의 파이프라인 스테이지로 나눈 후, 병렬로 연산하여 동작 속도를 높인다 [2-7]. 일반적으로, SHA-1은 라운드별로 연산기 구성이 다르기 때문에, 각각의 라운드를 스테이지로 구분한 4단 파이프라인 구조를 주로 사용한다. 이 경우 각 스테이지는 20번의 단계연산을 포함하며, 하나의 단계연산 혹은 언폴딩된 두 개 이상의 단계연산을 반복 실행을 허용한다. 그러나, 단계연산 혹은 언폴딩된 단계연산의 연산 지연은 라운드 연산 지연 시간의 정확한 약수가 아니기 때문에 파이프라인 구성시 성능 오버헤드가 발생된다.

이러한 문제를 해결하기 위해, 본 논문에서 제안한 SHA-1 파이프라인 구조는 단계연산 단위가 아닌 임계지연의 크기를 기준으로 파이프라인 스테이지를 구성하였다. 즉, 각 스테이지의 구성 회로는 라운드 연산의 전체 데이터패스를 반복 허용 횟수로 나누어 얻어진 연산 지연과 동일 혹은 가장 유사한 연산 회로를 포함한다. 따라서, 기존 단계연산에 기반한 파이프라인 구조와 달리 파이프라인 오버헤드가 적어짐으로써 성능이 향상된다.

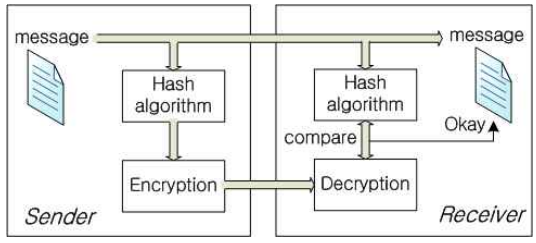
본 논문의 구성은 다음과 같다. 2장은 본 논문에서 제안한 SHA-1 해쉬 알고리즘과 언폴딩 및 파이프라인 기법에 대해 설명한다. 3장은 SHA-1 해쉬 알고리즘의 최대임계지연 분석과 제안된 SHA-1 파이프라인 구조를 설명한다. 4장은 제안된 SHA-1 프로세서의 회로 크기 및 성능에 대한 실험 결과를 제시하고, 5장에서 결론을 맺는다.

2. SHA-1 해쉬 및 성능 향상 기법

2.1. SHA-1 해쉬 알고리즘

SHA-1 해쉬 알고리즘은 최대 2^{64}-1 비트의 다양한 길이의 입력 메시지를 160 비트의 고정 비트열로 매핑한다. 해쉬 알고리즘을 사용하면 어떤 입력값이 주어졌을 때 같은 계산 결과를 얻을 수 있는 다른 입력값을 찾는 것이 불가능하다. 따라서, 데이터 전송 도중에 타인에 의하여 데이터가 변조되었는지를 쉽게 판단할 수 있기 때문에 전자서명 또는 중요 정보의 무결성 확인에 주로 사용된다.

전자서명의 경우 그림 1에 나타낸 것처럼 구현된다. 우선 전송자는 해쉬 함수를 이용하여 전송할 메시지의 해쉬값을 계산한다. 계산된 해쉬값은 비밀키를 사용하여 암호화하여 서명을 생성한 후, 메시지와 서명을 수신자에게 전송한다. 수신자는 수신된 메시지의 해쉬값을 계산한다. 동시에, 공개키를 이용하여 수신된 서명을 복호화하여 수신된 해쉬값을 복원한다. 수신된 메시지의 해쉬값과 수신된 해쉬값이 서로 동일하면 정상적인 수신으로, 그렇지 않으면 변조된 데이터로 판단한다.

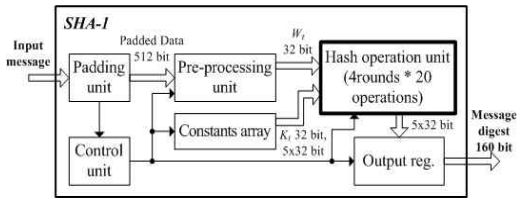


(그림 1) 전자서명을 통한 메시지 변조 확인

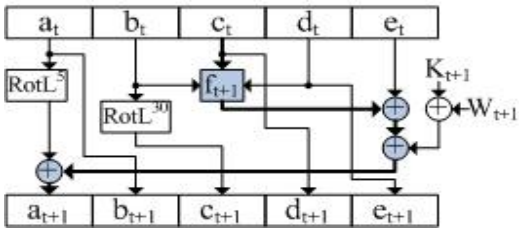
SHA-1 해쉬 연산을 위한 블록도는 그림 2와 같다. 최대 2^{64}-1 비트의 입력 메시지는 512 비트의 메시지 블록 단위로 구성된다. 각 메시지 블럭은 4 라운드 연산을 수행하며, 각 라운드는 20개의 단계연산을 포함한다. 그림 2에 나타낸 것처럼 각 라운드별로 스크램블링 상수, K_t와 비선형 연산 F_t가 변화한다. 프리프로세싱 (Pre-processing) 유닛을 통해 해쉬 연산에 사용되는 초기값을 세팅한다. 그림 2의 해쉬 연산 유닛 (Hash operation unit)을 통해 512비트의 메시지 블록별로 해쉬 연산을 수행한다.

해쉬 단계연산은 식 (1)과 같이 표현되며 이는 그림 3에 나타낸 단계연산을 80번 반복 수행하여 메시지 다이제스트를 출력한다. 식 (1)의 S_5와 S_{30}은 5비트와 30비트 왼쪽 순환 쉬프트 연산을 나타낸다. f_t 함수는 비선형 함수로 해당 라운드에 따라 변화한다. 단계연산의 임계경로는 짙은 색으로 표시되었고, 최대임계지연은 3개의 덧셈기와 1개의 비선형 함수의 합이 된다.

$$
\begin{aligned}
a_{t+1} &= S_5(a_t) + f_{t+1}(b_t, c_t, d_t) + e_t + w_t + k_t \quad (1)\\
b_{t+1} &= a_t\\
c_{t+1} &= S_{30}(b_t)\\
d_{t+1} &= c_t\\
e_{t+1} &= d_t
\end{aligned}
$$



(그림 2) 일반 SHA-1 블록도

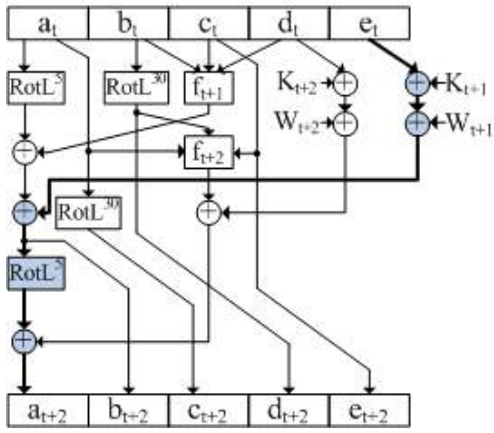


(그림 3) SHA-1 단계연산 블록

2.2. 언폴딩 알고리즘

언폴딩은 반복적인 연산을 수행할 때, 하나의 클럭 사이클 동안 2개 이상의 단계연산을 수행하는 기법을 의미한다. 2개 이상의 단계연산을 결합하여 새로운 동시 실행 가능한 데이터패스를 추가하여 성능 향상을 꾀한다. 그림 3에 나타낸 일반적인 SHA-1의 단계연산은 3개의 덧셈기와 1개의 비선형함수, f_t를 통과하는 시간이 최대임계지연시간이 되고, 이를 80 클럭동안 반복 실행하게 된다. 따라서, 메시지 다이제스트를 출력하기 위해서는 240개의 덧셈기와 80개의 비선형함수를 통과하는 시간이 전체 지연시간이 된다.

그림 4에 나타낸 것처럼 두 개의 단계연산을 하나의 연산으로 통합시 최대임계지연이 4개의 덧셈기와 1개의 비선형함수, f_t를 통과하는 시간으로 변화하며, 단지 40개의 클럭이 필요하며, 최종적으로 160개의 덧셈기와 40개의 비선형함수를 통과하는 시간으로 레이턴시가 감소하고, 결과적으로 처리 속도가 향상된다.

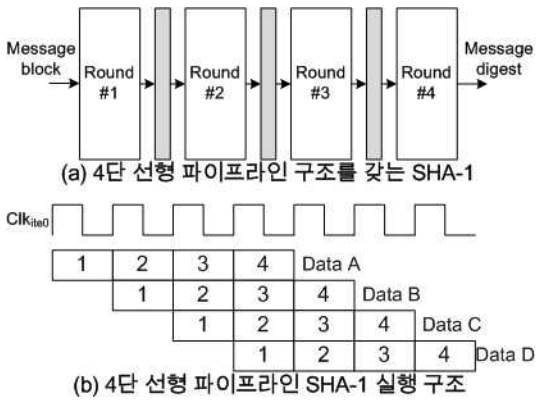


(그림 4) 2-언폴딩 SHA-1 해쉬 연산 블록

2.3 파이프라인 구조

언폴딩을 통한 병렬성 증대로 SHA-1 성능향상이 가능하나, 더 높은 처리량을 얻기 위해서 파이프라인 구조를 추가적으로 채택한다. 일반적으로 그림 5에 나타낸 것처럼 SHA-1 설계는 4개의 라운드를 파이프라인 스테이지로 선형적으로 구성한 4단 파이프라인 구조를 채택한다. 표 1에 나타낸 것처럼 라운드연산별로 4개

의 서로 다른 비선형함수, f_t와 상수, K_t를 갖기 때문에 제어 구조를 단순하게 할 수 있다는 장점을 갖는다. 파이프라인 구조를 갖는 SHA-1은 매 클럭마다 연속적인 메시지 블록이 입력되며, 4개의 파이프라인 스테이지들은 동시에 실행된다. 따라서, 파이프라인 구조를 갖지 않는 SHA-1에 비해 4배의 성능 향상을 갖는다.



(그림 5) 4단 선형 파이프라인 구조를 갖는 SHA-1 및 파이프라인 실행 스킴

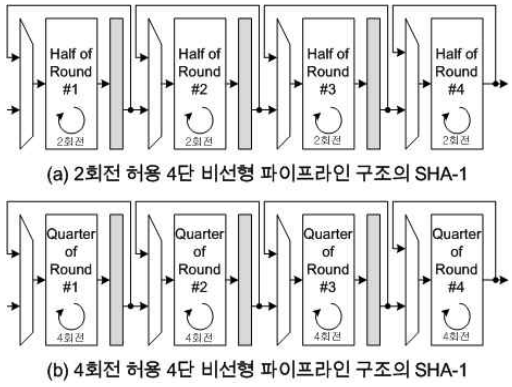
<표 1> 라운드별 비선형함수, f_t와 상수, K_t

t	f_t(x,y,z)	K_t
$0 \le t \le 19$	Ch(x,y,z)=(x∧y)⊕(x∧z)	5a827999
$20 \le t \le 39$	Parity(x,y,z)=x⊕y⊕z	6ed9eba1
$40 \le t \le 59$	Maj(x,y,z)=(x∧y)⊕(x∧z)⊕(y∧z)	8f1bbcdc
$60 \le t \le 79$	Parity(x,y,z)=x⊕y⊕z	ca62c1d6

3. 제안된 SHA-1 파이프라인 구조

일반적인 파이프라인은 그림 5에 나타낸 것처럼 전체 데이터패스의 실행 순서에 따라 파이프라인 스테이지들을 직렬 연결한 선형 파이프라인 구조를 갖는다. 그러나, VLIW (very long instruction word) 프로세서는 피드백 연결을 갖는 비선형 파이프라인 구조를 갖는다. 본 논문은 그림 6에 나타낸 것처럼 파이프라인 스테이지의 반복 연산을 허용하는 피드백 연결을 갖는다.

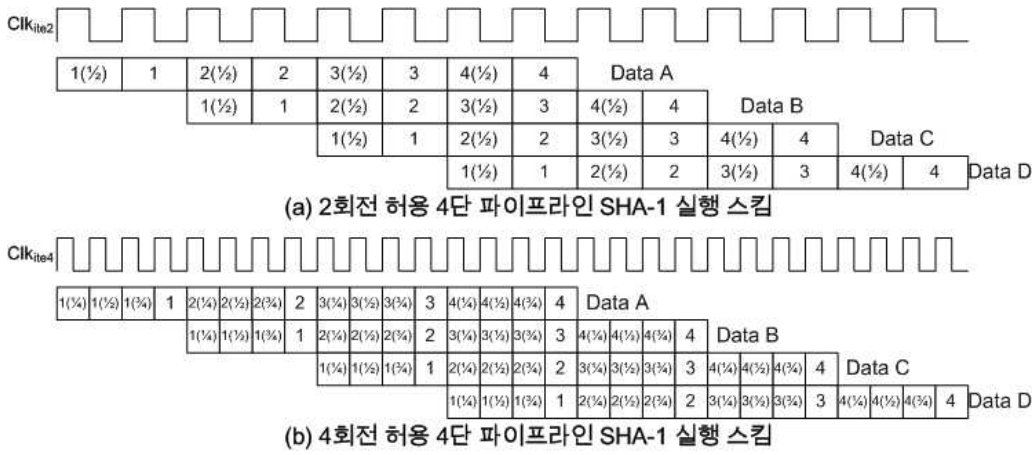
제안된 비선형 파이프라인 구조의 SHA-1 회로는 그림 6과 같이 구성된다. 파이프라인 스테이지 수는 기존 라운드별 파이프라인으로 구성한 SHA-1 회로와 같이 4개로 구성된다. 제안한 비선형 파이프라인은 각 스테이지 출력을 입력으로 회귀시켜 반복 실행을 허용한다. 즉, 그림 6(a)에 나타낸 것처럼 각 스테이지가 2회전을 허용할 경우, 해당 스테이지는 20번의 단계연산을 포함하는 대신 그 절반의 지연시간을 갖는 연산 회로로 구성된다. 마찬가지로 그림 6(b)에 나타낸 것처럼 4회전을 허용하는 파이프라인은 전체 지연의 4분의 1의 지연을 갖는 연산 회로만을 포함한다.



(그림 6) 회전을 허용하는 비선형 파이프라인 구조의 SHA-1 회로

회전 연산을 허용하는 비선형 파이프라인 SHA-1은 동일한 파이프라인 스테이지를 갖는 선형 파이프라인 구조에 비해 컴팩트한 회로 설계가 가능하다. 즉, 파이프라인 구조를 갖는 SHA-1의 회로 크기는 허용되는 회전 수에 비례하여 감소한다. 그림 5에 나타낸 기존 라운드별 파이프라인 구조를 갖는 SHA-1은 각 파이프라인 스테이지가 하나의 라운드로 구성되어 20번의 단계연산을 포함한다. 그러나, 2회전을 허용하는 비선형 파이프라인의 각 스테이지는 그 절반의 연산만을 포함하기 때문에 회로크기는 약 절반으로 감소된다.

Y. Lee는 임계지연에 따른 SHA-1 알고리즘의 최대 동작 속도 결정 방법을 제안하였다 [5]. 이에 따라, 파이프라인 스테이지에 포함된 단계연산의 수에 따른 최대임계경로의 크기는 표 2와 같이 나타낸다. 여기서, A는 덧셈기의 레이턴시이고, F는 비선형함수의 레이턴시를 나타낸다. 로테이션 연산은 지연시간이 적어



(그림 7) 회전을 허용하는 비선형 파이프라인 구조를 갖는 SHA-1 실행 스킴

일반적으로 무시된다.

일반적으로, 파이프라인 구조의 SHA-1은 각 스테이지가 하나의 라운드 연산으로 구성된다. 라운드에 포함된 20번의 단계연산을 모두 언폴딩하여 스테이지로 구성시, 최대임계경로의 크기는 22A+10F가 되고, 한 번의 사이클로 동작이 완료된다. 이의 파이프라인 통과지연은 22A+10F로 파이프라인 구성에 따른 오버헤드가 없다. 그러나, 회로 크기가 커지기 때문에 대부분 SHA-1 파이프라인은 표 3과 같이 하나의 단계연산 혹은 언폴딩된 두 개의 단계연산을 스테이지로 구성하고 20회 및 10회 반복 실행한다. 이 때, 각 스테이지 통과지연은 각각 40A+10F 및 60A+20F가 되어, 각각 18A와 38A+10F의 파이프라인 오버헤드가 발생된다.

이러한 문제를 해결하기 위하여 본 논문은 표 4에 나타낸 것처럼 단계연산 단위가 아닌 스테이지의 임계경로의 크기에 따라 파이프라인 스테이지에 포함된 연산을 나누어 파이프라인을 구성하였다. 각각의 파이프라인 스테이지에 포함된 연산은 라운드 연산의 전체 지연을 회전 수로 나누어 파이프라인 구성시 오버헤드를 최소화하였다.

<표 2> 파이프라인 스테이지에 포함된 단계연산 수, n에 따른 최대임계경로의 크기.

언폴딩계수	20	10	8	5	4	2
N_{add}	22A	12A	10A	7A	6A	4A
N_{non}	10F	5F	3F	2F	1F	1F

<표 3> 파이프라인 스테이지에 포함된 단계연산 수, n에 따른 최대임계경로의 크기.

언폴딩 계수	스테이지 임계경로	사이클 수	스테이지 통과지연	오버헤드
20	22A+10F	1	22A+10F	0
10	12A+5F	2	24A+10F	2A
5	7A+2F	4	28A+8F	6A-2F
4	6A+1F	5	30A+5F	8A-5F
2	4A+1F	10	40A+10F	18A
1	3A+1F	20	60A+20F	38A+10F

<표 4> 제안된 SHA-1 파이프라인 구조의 스테이지 통과지연 및 이에 따른 오버헤드 비교

언폴딩 계수	스테이지 임계경로	사이클 수	스테이지 통과지연	오버헤드
10	11A+5F	2	22A+10F	0
5	6A+2F	4	24A+8F	2A-2F
4	5A+2F	5	25A+10F	3A

2회전, 4회전 및 5회전을 허용하는 제안된 비선형 파이프라인 구조의 SHA-1의 최대 파이프라인 오버헤드는 3A로 기존 단계연산을 기준한 파이프라인 구성시 동작속도가 크게 향상된다. 이는 기존 방식에 비해 높은 동작 클럭으로 구동되기 때문이다. 클럭 주기는 해당 데이터패스의 최대임계경로의 길이에 따라 결정

된다. 제안된 비선형 2회전 및 4회전 허용 4단 파이프라인의 클럭 주파수는 각각 1/(11A+5F), 그리고 1/(6A+2F)가 된다. 하나의 메시지 블록을 처리하기 위해서는 4개의 라운드 연산을 수행하기 때문에 제안된 2회전 및 4회전을 허용하는 비선형 파이프라인은 그림 7에 나타낸 것처럼 각각 8개 및 16개의 클럭이 요구된다. 결과적으로, 하나의 메시지 블록을 처리하는 최대 지연시간은 8×(11A+5F) 및 16×(6A+2F)이 되고, 전체 파이프라인 오버헤드는 0 및 8A-8F가 된다. 따라서, 기존 단계연산에 따른 4단 파이프라인 구성을 갖는 SHA-1의 오버헤드에 비해 큰 폭으로 감소된다.

4. 시뮬레이션 결과

제안된 비선형 파이프라인 구조를 갖는 SHA-1 회로는 Verilog HDL (hardware description language)를 이용하여 RTL (register transfer level) 레벨에서 합성하였고, Xilinx FPGA (field programmable gate array)를 이용하여 동작을 검증하였다. 제안된 SHA-1 설계는 표준에 정의된 테스트 메시지 셋을 이용하여 동작을 완벽하게 검증하였다. 또한, 공정한 성능 평가를 위해 Xilinx Vertex 2 FPGA, XC2V1000을 이용하여 다른 SHA-1 구조와 성능 및 회로 크기를 비교하였다.

본 논문에서 제안한 비선형 파이프라인은 회전수에 따라 각 스테이지별로 포함되는 연산의 최대임계지연의 크기를 최대한 동일한게 함으로써, 파이프라인 오버헤드를 줄였다.

첫 번째 실험은 기존 언폴딩계수에 따른 파이프라인 스테이지의 회로 크기와 동작 속도 분석을 수행하였고, 그 결과는 표 3과 같다. 언폴딩된 단계연산의 회로 크기는 언폴딩계수가 증가함에 따라 커진다. 일반적으로 파이프라인 구조에 사용되는 하나의 단계연산 및 2 언폴딩 단계연산은 3A+F 및 4A+F의 최대임계경로를 갖고, 115MHz와 80MHz에서 동작한다. 반면 본 논문에서 제안한 2회전 및 4회전 비선형파이프라인에 사용되는 5 언폴딩 및 10 언폴딩 단계연산은 7A+2F 및 12A+5F로 55MHz 및 28MHz의 최대동작주파수로 구성된다. 회로 크기의 경우, 기본 단계연산은 114개의 슬라이스로 구성됨에 반해, 2 언폴딩은 8.8% 커진 124 슬라이스로 구성된다.

첫 번째 실험과 동일한 환경으로 임계경로의 크기에 따라 파이프라인 스테이지를 구성한 후, 회전 수에 따라 회로 크기 및 성능 비교를 수행하였고, 결과는 표 6과 같다. 제안된 파이프라인 구조는 회전 수가 2, 4 그리고 5일 때 최대임계지연은 각각 33.6ns, 17.4ns, 그리고 14.9ns로 나타났고, 30MHz, 57MHz, 그리고 71MHz의 클럭주파수로 구동되었다. 따라서, 표 5에 나타낸 기존 단계연산을 기준한 SHA-1 파이프라인 구조에 비해 2회전을 허용하는 파이프라인은 1.07배, 4회전 및 5회전을 허용하는 파이프라인 구조는 1.14배 및 1.22배 동작 속도가 향상되었다. 또한, 회로 크기면에서 기존 단계연산을 기준한 SHA-1 파이프라인 구조에 비해 5회전 허용시 최대 18% 증가하였고, 2회전 허용시 최소 9.8% 증가하였다. 따라서, 회로 크기 증가율에 비해 성능 향상율이 높음을 증명하였다.

각각의 파이프라인 스테이지뿐만 아니라, 4단 파이프라인 구조를 갖는 SHA-1 전체의 성능 평가를 수행하였고, 이를 통해 회로 크기 및 성능 비교 결과는 표 7에 나타내었다. 표 7은 허용 회전수가 2와 4일 때, 본 논문에서 제안한 SHA-1 4단 비선형파이프라인 구조의 최대동작지연시간과 회로크기를 비교 결과를 나타낸다. 제안된 최대임계경로에 의한 SHA-1 파이프라인 구성은 20개의 단계연산을 모두 언폴딩한 라운드를 하나의 파이프라인 스테이지로 구성했을 때 회로 크기는 12,414 슬라이스 그리고 최대 처리 속도는 7.53Gbps를 나타낸다. 반면 제안된 2회전 및 4회전을 허용하는 SHA-1 파이프라인 구조는 7.48Gbps 및 7.04Gbps의 성능을 갖는다. 파이프라인 스테이지의 지연시간의 밸런스를 맞춤으로써, 각각 1.0% 및 7.5%의 성능 감소로 회로 크기는 각각 38.7% 및 65.2% 감소되었다. 결론적으로, 최대한 성능 감소를 줄이면서, 큰 폭의 회로 크기 감소가 가능함을 증명하였다.

표 8은 해쉬단계연산 및 언폴딩된 단계연산 단위로 파이프라인 스테이지를 구성한 기존 SHA-1 구조와의 비교를 나타낸다. 제안된 파이프라인 구조를 갖는 SHA-1은 4단 파이프라인을 갖는 기존 구조에 비해 회로 크기는 최대 4.8배에서 최소 1.8배 증가한다. 반면, 동작 속도는 최대 5.7배에서 최소 2.5배 증가한다. 특히, 회로크기에 따른 동작속도 비율에서 2회전 및 4회전을 허용하는 파이프라인 SHA-1이 0.99 및 1.62로 기존

구조에 비해 우월함을 증명하였다.

<표 5> 언폴딩 계수에 따른 파이프라인 스테이지의 회로 크기 및 동작 속도

언폴딩 계수	최대임계 경로	최대임계 지연(ns)	최대클럭 주파수(MHz)	회로크기 (Slices)
20	22A+10F	67.1	15MHz	291
10	12A+5F	36.1	28MHz	215
5	7A+2F	19.9	50MHz	181
4	6A+2F	17.4	58MHz	164
2	4A+F	11.2	89MHz	124
1	3A+F	8.7	115MHz	114

<표 6> 언폴딩 계수에 따른 파이프라인 스테이지의 회로 크기 및 동작 속도

허용 회전수	최대임계 경로	최대임계 지연(ns)	최대클럭 주파수(MHz)	회로크기 (Slices)
1	22A+10F	67.1	15MHz	291
2	11A+5F	33.6	30MHz	236
4	6A+2F	17.4	57MHz	205
5	5A+2F	14.9	71MHz	194

<표 7> 언폴딩 계수에 따른 파이프라인 스테이지별 회로 크기 및 최대동작지연시간 비교.

허용회전수	1	2	4
회로크기 (slices)	12,412	7,612	4,325
최대지연시간 (ns)	68.1	34.2	18.2
단일 메시지 블록 처리지연시간 (ns)	272.4	273.6	291.2
클럭주파수 (MHz)	14.7	29.2	55.0
최대 성능(Gbps)	7.53	7.48	7.04

<표 8> 기존 4단 파이프라인 구조를 갖는 SHA-1과 성능 및 회로 크기 비교

	Devices	Slices	Throughput (Gbps)	Throughput /slices
[2]	Xilinx V100	1,578	1.7	1.08
[3]	Xilinx 2V500	2,245	1.3	0.58
[4]	Xilinx V150	N.A	2.8	N.A
[5]	Virtex-Ⅱ	4,258	2.5	0.59
[8]	Virtex-Ⅱ (non-pipe)	2,894	5.9	2.04
Prop.	Virtex-Ⅱ	7,612	7.5 (2회전)	0.99
		4,325	7.0 (4회전)	1.62

5. 결론

본 논문은 SHA-1 알고리즘의 해쉬 단계연산에 따른 최대 임계 경로를 분석하고, 이의 레이턴시를 분석하였다. 기존 파이프라인 구조를 갖는 SHA-1 회로는 단계연산 자체 혹은 2개 이상의 단계 연산의 언폴딩 회로를 토대로 회전을 허용하는 파이프라인 구조를 갖는다. 기존 구조는 파이프라인 구성시 각 스테이지가 최적의 임계지연과 차이를 갖기 때문에 회전 수가 많아질수록, 파이프라인 오버헤드가 누적되는 단점을 갖는다. 본 논문은 하나의 라운드를 하나의 파이프라인 스테이지로 구성하고, 반복 실행을 허용하는 점에서 기존 구조와 동일하다. 반면, 각 파이프라인 스테이지는 단계연산이 아닌 라운드별 최대임계지연을 기준으로 회전 수에 따라 최적화함으로써, 파이프라인 구성에 따른 오버헤드를 최소화하였다. 이를 통해, 기존 4단 파이프라인 구조를 갖는 SHA-1과 비교하여 회로 크기는 1.8배에서 4.8배까지 증가하는 반면, 동작 속도는 2.5배에서 최대 5.7배로 증가한다. 결론적으로, 회로크기에 따른 동작속도 비율에서 2회전 및 4회전을 허용하는 파이프라인 SHA-1이 0.99 및 1.62로 기존 구조에 비해 우월함을 증명하였다. 제안된 구조는 SHA-1 뿐만 아니라 반복 실행을 필수적으로 수행하는 다양한 암호회로에 적용 가능할 것으로 기대된다.

참고문헌

[1] L. Jiang, Y. Wang, Q. Zhao, Y. Shao and X. Zhao, "Ultra high throughput architectures for SHA-1 Hash algorithm on FPGA," Proc. of CiSE *2009*, pp. 1-4, 2009.

[2] N. Sklavos, E. Alexopoulos and O. Koufopavlou, "Networking data integrity: High speed architecture and hardware implementation," The Int'l Arab J. of Information Technology, vol. 1, no, 6, pp. 54-59, July 2003.

[3] N. Sklavos, E. Alexopoulos and O. Koufopavlou, "An ultra high speed architecture for VLSI implementation of Hash functions," Proc. of ICECS 2003, pp. 990-993, 2003.

[4] H. Michail, A. Kakarountas, O. Koufopavlou and C. Goutis, "A low-power and high-throughput implementation of the SHA-1 Hash function," Proc. of ISCAS 2005, pp. 23-26, 2005.

[5] Y. Lee, H. Chan and I. Verbauwhede, "Throughput optimized SHA-1 architecture using unfolding transformation, Proc. of ASAP 2006, pp. 354-359, 2006.

[6] S. Suhaili and T. Watanabe, "High throughput evaluation of SHA-1 implementation using unfolding transformation," ARPN J. of Engineering and Applied Sciences, vol. 11, no. 5, pp. 3350-3355, March 2016.

[7] H. Michail, G. Arthanasiou, G. Theodoridis, A. Gregoriades and C. Goutis, "Design and Implementation of totally-self checking SHA-1 and SHA-256 hash functions's architecture," Microprocessors and Microsystems, vol. 45, part B, pp. 227-240, Sep. 2016.

[8] E. Lee, J. Lee, I. Park and K. Cho, "Implementation of high-speed SHA-1 architecture," IEICE ELEX, vol. 6, pp. 1174-1179, Aug. 2009.

[著者紹介]

이 제 훈 (Je-Hoon Lee)
1998년 8월 공학사 충북대학교 정보통신공학과
2001년 2월 공학석사 충북대학교 정보통신공학과 통신회로및시스템공학
2005년 2월 공학박사 충북대학교 정보통신공학과 통신회로및시스템공학
2005년 - 2006년 USC 방문 연구원
2006년 - 2009년 충북대학교 초빙조교수
2009년 - 현재 강원대학교 전자정보통신공학부 부교수
관심분야 : 회로설계, 헬스케어, IoT
email : jehoon.lee@kangwon.ac.kr

최 규 만 (Gyu-Man Choi)
1981년 2월 부산대학교 물리학과 이학사
1983년 2월 부산대학교 대학원 이학석사
1991년 2월 경북대학교 전자공학과 공학박사
1983년 - 1989년 삼성SDI 종합연구소 선임연구원
1989년 - 현재 가톨릭관동대학교 전자공학과 교수
관심분야 :전자재료, 센서디스플레이
email : kmchoi@cku.ac.kr