(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(51) 。Int. Cl. [7]
H04L 9/06

(45) 공고일자 2003년03월26일
(11) 등록번호 10-0377175
(24) 등록일자 2003년03월10일

(21) 출원번호 10-2000-0031247
(22) 출원일자 2000년06월08일

(65) 공개번호 특2001-0111120
(43) 공개일자 2001년12월17일

(73) 특허권자 주식회사 하이닉스반도체
경기 이천시 부발읍 아미리 산136-1

(72) 발명자 임영원
경기도이천시대월면현대전자사원아파트106-1302

(74) 대리인 특허법인 신성
최종식
정지원

심사관 : 정재헌

(54) 데이터 암호화 표준 알고리즘을 이용한 암호화 장치

요약

본 발명은 S-Box 치환부를 2-포트의 입력을 가진 장치로 구현한 데이터 암호화 표준 알고리즘을 이용한 암호화 장치에 관한 것으로, 억세스(Access) 시간이 두 배 빠른 기억장치를 사용하여 주어진 시간에 데이터를 두 번 억세스하기 때문에 데이터 컨텐션(Contention)을 제거하여 면적을 최소화할 수 있는 암호화 장치를 제공하기 위한 것으로, 이를 위해 본 발명은, 데이터 초기치환부, 제1 클럭, 제2 클럭 및 제3 클럭을 이용하는 n단 (n은 짝수) 파이프라인 구조를 갖는 데이터 변형부 및 데이터 역초기치환부를 포함하고, 데이터 암호화 표준 알고리즘을 사용하여 평문 텍스트 블록의 암호화를 수행하기 위한 암호화 장치에 있어서, n/3 개의 48비트의 데이터 중에 하나를 선택하기 위한 멀티플렉서; 상기 멀티플렉서에서 출력된 48비트 중에서 6비트의 어드레스 8개를 각각 입력받아 4비트의 데이터 8 개를 각각 출력하는 8 개의 S-Box를 포함하는 S-Box 치환부; 상기 32비트의 데이터를 n/3개의 출력으로 분배하는 디멀티플렉서; 및 제4클럭 및 제5클럭에 따라 상기 멀티플렉서 및 상기 디멀티플렉서를 제어하는 제어기를 포함하고, 상기 제4 클럭 및 상기 제5 클럭은 상기 제1 클럭, 상기 제2 클럭 및 상기 제3 클럭보다 1/3 주기만큼 빠르며, 상기 제4클럭과 제5클럭은 서로 반전된 신호임을 특징으로 하는 암호화 장치를 제공한다.

대표도

도 7

색인어

멀티플렉서, 제어기, S-Box, 디멀티플렉서

명세서

도면의 간단한 설명

도1은 일반적인 DES 아키텍쳐의 사이퍼 함수와 S-Box 치환부의 상세한 구성도,

도2는 처리능력을 높이는 효과가 있는 본 발명이 적용되는 3개의 위상을 가지는 클럭을 이용한 6단의 파이프라인 구조의 DES 아키텍쳐를 나타내는 블럭도,

도3은 도2의 6단 파이프라인 DES 아키텍쳐의 동작 순서를 나타내는 타이밍도,

도4는 도2의 6단 파이프라인 DES 아키텍쳐의 파이프라인 동작 순서를 나타내는 타이밍도,

도5는 도2의 6단 파이프라인 DES 아키텍쳐의 파이프라인을 사용하지 않은 경우와 파이프라인을 사용한 경우에 사이퍼 함수가 연산되는 순서도

도6은 종래기술의 단일 포트 S-Box 치환부의 상세한 블럭도,

도7은 본 발명의 2-포트 S-Box치환부의 구현방식을 나타낸 상세한 블럭도,

도8은 종래 방식의 단일 포트 S-Box 치환부와 본 발명의 2-포트 S-Box 치환부의 동작을 나타내는 타이밍도.

* 도면의 주요 부분에 대한 부호의 설명 *

710 : 멀티플렉서 720 : S-Box

730 : 디멀티플렉서 740 : 제어기

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

본 발명은 암호화 장치에 관한 것으로, 특히 데이터 암호화 표준 알고리즘을 이용한 암호화 장치에 관한 것이다.

일반적으로 데이터 암호화 표준(DES : Data Encryption Standard, 이하 DES라 칭함) 알고리즘은 가장 널리 쓰이고 있는 암2호화 방식으로 네트워킹 사용이 증가함에 따라 그 중요성을 더해 가고 있다. 특히, 보안 인터넷 응용이나 원격 접근 서버나 케이블 모뎀과 위성용 모뎀 등의 분야에서 많이 이용되고 있다.

DES는 기본적으로 64비트 블럭의 입력 및 출력을 가지는 64비트 블럭 암호이며, 64비트의 키 블럭 중 56비트가 암호화 및 복호화에 사용되고, 나머지 8비트는 패리티 검사용으로 사용된다. 또한. 64비트의 평문(Plain Text) 블럭과 56비트의 키(Key)를 입력으로 해서 64비트의 암호문(Ciper Text) 블럭을 출력하는 암호화 장치이다.

DES를 실현시키는 중요한 기법은 치환(P-Box), 대치(S-Box) 그리고 보조키(Subkey)를 발생시키는 키 스케쥴러(Key scheduler) 등이 있다.

데이터 암호화부의 내부는 16라운드의 반복연산을 수행하는 형태로 되어 있고 입력부의 초기 치환(IP)와 출력부의 역초기 치환(IP^{-1})으로 구성되어 있다.

도1는 일반적인 DES 아키텍쳐의 사이퍼 함수와 S-Box 치환부의 상세한 구성도이다.

상기 도1를 참조하면, 사이퍼 함수 f는 32비트의 텍스트 블럭을 저장하고 있는 오른쪽레지스터로부터 32비트의 데이터(R(i-1))를 입력받아 48비트의 데이터로 확장 치환하는 확장치환부(110)와, 확장치환부의 48비트의 데이터를 입력받고 키 스케쥴러로부터의 보조키(K_i)를 입력받아 배타적 논리합 연산을 수행하는 배타적 논리합 연산부(120)와, 배타적 논리합 연산부(120)로부터의 48비트의 데이터를 32비트의 데이터로 대치 치환하는 S-Box 치환부(130)와, S-Box 치환부(130)의 32비트의 데이터를 복사 치환하는 P-Box 치환부(140)와, P-Box 치환부의 32비트의 데이터와 왼쪽레지스터에 저장되어 있는 32비트의 데이터(L(i+1))를 입력받아 배타적 논리합하는 배타적 논리합 연산부(150)를 구비한다.

키 스케쥴러는 각각 두 28비트 블럭을 56비트 키로부터 입력받아 쉬프트하는 두 개의 쉬프트부(160, 170)와, 쉬프트부들로부터 수신한 두 블럭을 하나의 보조키로 압축하여 치환하는 압축치환부(180)를 구비한다.

구체적으로, S-Box 치환부는 48비트의 입력을 받아서 32비트의 출력을 생성하는 8 개의 S-Box로 구성되어 있다. 즉, 48비트의 데이터는 8 개의 6비트 데이터로 분할되어 8 개의 S-Box에 입력된다. 이 8 개의 S-Box들은4비트 데이터를 출력함으로써 48 비트를 32비트로 변환한다. S-Box 치환부(320)는 룩-업(Look-up) 방식으로 실행됨으로써 프로그램가능 논리 어레이(PLA)나 롬(ROM)과 같은 기억장치를 필요로 한다. 6비트의 입력에 대하여 4비트를 출력하기 때문에 각 S-Box는 64 × 4 의 기억 용량이 필요하며 전체적으로 8개의 S-Box로 구성되어 있으므로 8 × 64 × 4의 기억장치가 필요하다. 따라서 전체적으로 칩에서 차지하는 면적이 상대적으로 크다.

도2는 처리능력을 높이는 효과가 있는 본 발명이 적용되는 3개의 위상을 가지는 클럭을 이용한 6단의 파이프라인 구조의 DES 아키텍쳐를 나타내는 블럭도이다.

도2를 참조하면, 본 발명의 DES 알고리즘은 초기 치환부를 거친 64비트의 평문(Plain Text) 블럭을 32비트의 두 블럭으로 나누어 a_0와 b_0를 제1클럭과 제2클럭을 사용하여 제1왼쪽레지스터(A0)(350)와 제1오른쪽레지스터(B0)(200)에 저장한 다. 그런 후, 키 스케쥴러로부터 생성된 보조키($K_{(i)}$)를 입력받아 제1오른쪽레지스터(B0)(200)로부터의 32비트의 데이터를 사이퍼 함수 f(210)에 의해서 암호화 변형하며, 사이퍼 함수 f에 의해 변형된 제1왼쪽레지스터(A0)(350)의 32비트와 배타적 논리합 연산부(220)에서 배타적 논리합 연산을 수행한다. 또한, 배타적 논리합 연산부(220)의 32비트 데이터를 제3클럭(CLK3)을 사용하여 제2왼쪽레지스터(C0)(230)에 저장한다.그런 후, 키스케쥴러에서 생성된 보조키($K_{(i+1)}$)를 입력받아 제2왼쪽레지스터(C0)(230)에 저장되어 있는 32비트의 데이터를 사이퍼 함수 f(240)를 통하여 암호화하여 변형하며, 변형된 32비트의 데이터를 제1오른쪽레지스터(B0)(200)의 32비트와 배타적 논리합 연산부(250)에서 배타적 논리합 연산을 수행한다. 또한, 배타적 논리합 연산부(250)의 32비트 데이터를 제1클럭(CLK1)을 사용하여 제2오른쪽레지스터(A1)(260)에 저장한다.그런 다음, 보조키($K_{(i+2)}$)를 입력받아 제2오른쪽레지스터(A1)(260)에 저장되어 있는 32비트의 데이터를 사이퍼 함수 f(270)를 통하여 변형하며, 변형된 32비트의 데이터를 제2왼쪽레지스터(C0)(230)의 변형된 32비트와 배타적 논리합 연산부(280)에서 배타적 논리합 연산을 수행한다. 또한 배타적 논리합 연산부(280)의 32비트 데이터를 제2클럭(CLK2)을 사용하여 제3왼쪽레지스터(B1)(290)에 저장한다.보조키($K_{(i+3)}$)를 입력받아 제3왼쪽레지스터(B1)(290)에 저장되어 있는 32비트의 데이터를 사이퍼 함수 f(300)를 통하여 암호화함으로써 변형하며, 변형된 32비트의 데이터를 제2오른쪽레지스터(A1)(260)의 변형된 32비트와 배타적 논리합 연산부(310)에서 배타적 논리합 연산을 수행한다. 또한, 배타적 논리합 연산부(310)의 32비트 데이터를 제3클럭(CLK3)을 사용하여 제3오른쪽레지스터(C1)(320)에 저장한다.보조키($K_{(i+4)}$)를 입력받아 제3오른쪽레지스터(C1)(320)에 저장되어 있는 32비트의 데이터를 사이퍼 함수 f(330)를 통하여 변형하며, 변형된 32비트의 데이터를 제3왼쪽레지스터(B1)(290)의 변형된 32비트와 배타적 논리합 연산부(340)에서 배타적 논리합 연산을 수행한다. 또한 배타적 논리합 연산부(340)의 32비트 데이터를 제1클럭(CLK1)을 사용하여 제1왼쪽레지스터(A0)(350)에 저장한다.보조키($K_{(i+5)}$)를 입력받아 제1왼쪽레지스터(A0)(350)에 저장되어 있는 32비트의 데이터를 사이퍼 함수 f(360)를 통하여 암호화함으로써 변형하며, 변형된 32비트의 데이터를 제3오른쪽레지스터(C1)(320)의 변형된 32비트와 배타적 논리합 연산부(370)에서 배타적 논리합 연산을 수행한다.

마지막 라운드의 제3왼쪽레지스터(B1)(290)의 32비트가 블럭 b_{15}가 되며 마지막 라운드의 배타적 논리합 연산부(310)에서 출력된 32비트가 b_{16}이 된다.

제2클럭(CLK2)은 제1클럭(CLK1)을 1/3 주기만큼 지연시킨 것이고 제3클럭(CLK3)은 제2클럭(CLK2)를 1/3 주기만큼 지연시킨 것이다. 제1클럭(CLK1)이 상승할 때 레지스터 A0와 A1에 새로운 값이 저장되고, 제2클럭(CLK2)이 상승할 때 레지스터 B0와 B1에 새로운 값이 저장되고, 제3클럭(CLK3)이 상승할 때 레지스터 C0와 C1에 새로운 값이 저장된다.

도3는 도2의 6단 파이프라인 구조의 DES 아키텍쳐의 동작 순서를 나타내는 타이밍도이다.

도3을 참조하면, 32비트의 블럭 a_0와 b_0는 초기 치환을 거친 64비트의 평문 블럭이 32비트의 두블럭으로 나뉘어진다. a_0와 b_0 값은 t_0와 t_1에서 제1클럭(CLK1)과 제2클럭(CLK2)에 의해서 레지스터 A0와 B0에 각각 저장된다. t_1에서부터 b_1 값($b_1 = a_0 \oplus f(b_0, K_1)$)을 계산하기 시작해서 t_2에서 계산된 값을 레지스터 C0에 저장한다. 이 때 레지스터 A0에 입력된 값 a_0는 t_3까지 유지가 되어서 $t_1 - t_2$ 구간에서 b_1 값을 계산하는데 사용할 수 있고 b_1 값은 t_5까지 유지되어서 $t_2 - t_3$ 구간에서 b_2 값을 계산하는데 사용할 수 있다. 이 것은 서로 지연된 제1클럭(CLK1)과 제2클럭(CLK2)과 제3클럭(CLK3)에 의해서 레지스터 A0와 B0와 C0가 새로운 값을 저장하기 때문에 해결할 수 있다. b_2 ($b_2 = b_0 \oplus f(b_1, K_2)$)값은 $t_2 - t_3$ 구간에서 계산되어 t_4에서 제2클럭(CLK2)에 의해서 레지스터 B1에 저장된다. 이와같이 3 개의 위상을 갖는 클럭을 사용하여 레지스터에 저장된 값을 동시에 억세스(Access)하게 함으로써 b_1, b_2, ..., b_{16}을 계산하는데 걸리는 시간을 5.66 클럭 사이클로 줄일 수 있다.

일반적으로 주어진 키(Key)에 대하여 암호화하거나 해독화할 다수의 64비트의 평문이나 암호문 블럭이 연속적으로 입력되는 경우가 많다. 예를 들어 MCNS 케이블 모뎀에서 사용되는 암호화 방식은 맥 프레임(MAC Frame) 단위로 암호화를 수행하기 때문에 최대 1,518 바이트의 평문 블럭들을 동일한 키(Key)로 암호화를 수행하여야 한다. 즉 동일한

키(Key)로써 다수의 평문 블럭들에 대해서 16라운드의 DES 코아를 계산하여야만 한다. 이 경우 종래 기술이 가지고 있는 파이프라인 구조를 이용하여 처리능력비를 증가시킬 수 있다.

도4은 도2의 6단 파이프라인 구조의 DES 아키텍쳐의 파이프라인 동작 순서를 나타내는 타이밍도이다.

도4을 참조하면, 본 발명이 적용되는 6단 파이프라인 DES 구조는 파이프라인 구조를 이용하여 두개의 평문 블럭들을 5.66 클럭 사이클 동안 동시에 처리할 수 있음을 보여준다. 또한 도3에서 비어 있는 부분에 새로운 평문 블럭 c_0와 d_0를 t_3와 t_4에서 레지스터 A0와 B0에 입력함으로써 평문 블럭 b_i값들을 계산하는 동안 평문 블럭 d_i값들을 계산할 수 있음을 보여준다. 이 때 t_0-t_1, t_1-t_2, t_2-t_3.... 구간마다 새로운 평문 블럭 b_i와 d_i값을 암호화 하기위해 사이퍼 함수 f가 두 개씩 동시에 수행된다. 5.66 클럭 사이클 동안 처리할 수 있는 평문 블럭의 수는 두배로 증가시킬 수가 있다. 그러나 사이퍼 함수를 구성하는 S-Box들은 한 개씩 추가로 구현할 필요가 있는 단점이 발생한다.

사이퍼 함수 f는 롬(ROM)이나 프로그램가능 논리 어레이(PLA)와 같은 기억장치로 구현된 S-Box 치환부를 필요로 한다.

도5는 도2의 6단 파이프라인 구조의 DES 아키텍쳐의 파이프라인을 사용하지 않은 경우와 파이프라인을 사용한 경우에 사이퍼 함수가 연산되는 순서도이다.

도5를 참조하면, 한개의 64비트 평문 블럭을 암호화하는 경우 즉, 파이프라인을 사용하지 않은 경우에는 도2의 6개의 사이퍼 함수 f_A, f_B, f_C, f_D, f_E, f_F는 3개의 위상을 갖는 클럭에 의해서 시분할이 되어 계산되기 때문에 1 개의 S-Box 치환부만으로도 구현가능하다. 그러나 파이프라인을 사용하여 두개의 64비트의 평문 블럭을 동시에 암호화할 경우에 사이퍼 함수들의 두 집합 (f_A, f_B, f_C)와 (f_D, f_E, f_F)는 서로 시분할되지만 세 집합 (f_A,f_D)와 (f_B,f_E)와 (f_C,f_F)는 시분할이 되지 않고 동시에 계산되기 때문에 두 개의 S-Box 치환부가 필요하다.

도6은 종래기술의 단일 포트 S-Box 치환부의 상세한 블럭도이다.

도6을 참조하면, 종래기술은 S-Box 치환부 두 개를 사용하여 파이프라인 동작을 수행하도록 되어 있고, 개개의 S-Box 치환부는 48비트의 입력 데이터를 받아들여 32비트의 출력 데이터를 내보내는 8 개의 S-Box로 구성되어 있다. 각각의 S-Box는 64 × 4 의 롬(ROM)이나 프로그램가능 논리 어레이(PLA)로 구성되어 있고 6비트의 어드레스를 입력받아 4비트의 데이터를 출력하는 제1경로를 구비하고 있다. 두개의 S-Box 치환부에는 서로 다른 제1경로와 제2경로가 물리적으로 존재한다.

종래 기술은 상기와 같이 서로 다른 경로가 물리적으로 존재하여 S-Box 치환부를 구현하는데 필요한 기억장치를 동시에 억세스(Access)해야 하는 데이터 컨텐션(Data contention) 문제를 해결한다. 따라서 똑 같은 S-Box를 두개를 사용하므로 인하여 면적이 증가되는 단점이 있다.

발명이 이루고자 하는 기술적 과제

본 발명은 상기와 같은 종래기술의 문제점을 해결하기 위하여 안출된 것으로써, 데이터 컨텐션(Contention)을 제거하고 면적을 최소화할 수 있는 암호화 장치를 제공하는데 그 목적이 있다.

발명의 구성 및 작용

상기 본 발명의 목적을 달성하기 위하여 본 발명의 암호화 장치는, 데이터 초기치환부, 제1 클럭, 제2 클럭 및 제3 클럭을 이용하는 n단 (n은 짝수) 파이프라인 구조를 갖는 데이터 변형부 및 데이터 역초기치환부를 포함하고, 데이터 암호화 표준 알고리즘을 사용하여 평문 텍스트 블록의 암호화를 수행하기 위한 암호화 장치에 있어서, n/3 개의 48비트의

데이터 중에 하나를 선택하기 위한 멀티플렉서; 상기 멀티플렉서에서 출력된 48비트 중에서 6비트의 어드레스 8개를 각각 입력받아 4비트의 데이터 8 개를 각각 출력하는 8 개의 S-Box를 포함하는 S-Box 치환부; 상기 32비트의 데이터를 n/3개의 출력으로 분배하는 디멀티플렉서; 및 제4클럭 및 제5클럭에 따라 상기 멀티플렉서 및 상기 디멀티플렉서를 제어하는 제어기를 포함하고, 상기 제4 클럭 및 상기 제5 클럭은 상기 제1 클럭, 상기 제2 클럭 및 상기 제3 클럭보다 1/3 주기만큼 빠르며, 상기 제4클럭과 제5클럭은 서로 반전된 신호임을 특징으로 하는 암호화 장치를 제공한다.

이하, 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자가 본 발명의 기술적 사상을 용이하게 실시할 수 있을 정도로 상세히 설명하기 위하여, 본 발명의 가장 바람직한 실시예를 첨부한 도면을 참조하여 설명하기로 한다.

도7은 본 발명의 2-포트 S-Box 치환부의 구현방식을 나타낸 상세한 블럭도이다.

도7을 참조하면, 본 발명의 S-Box 치환부는 제어기의 제어를 받아 입력되는 두 개의 48비트의 데이터 중에 하나를 선택하는 멀티플렉서(710)와, 멀티플렉서(710)에서 출력된 48비트 중에서 6비트의 어드레스 8 개를 입력받아 4비트의 데이터 8 개를 출력하는 8 개의 S-Box(720)와, 제어기의 제어를 받아 상기 4비트의 데이터를 두개로 분배하는 디멀티플렉서(730)와, 제1클럭(CLK_A)과 제2클럭(CLK_B)를 입력받아 멀티플렉서(710)와 디멀티플렉서(730)를 제어하는 제어기(740)를 구비한다.

도8은 종래 방식의 단일 포트 S-Box 치환부와 본 발명의 2-포트 S-Box 치환부의 동작을 나타내는 타이밍도이다.

상기 도8을 참조하면, 본 발명에서는 제어기에 입력되는 두 배가 빠른 제1클럭(CLK_A)과 제2클럭(CLK_B)을 이용하여 롬(ROM)을 엑세스(Access)하는데 필요한 신호들을 발생시킨다. 각 시간 구간 $t_i - t_{i+1}$ 에서 제1경로(path1)과 제2경로(path2) 중의 한 경로를 선택하는 멀티플렉서에 의해 시분할된 제1경로(path1)과 제2경로(path2)가 개념적으로 존재하여 데이터 컨텐션(Data Contention) 문제를 해결한다. 즉, 제1클럭(CLK_A)이 '논리 하이'일 때 제1경로(path1)을 선택하여 b_i값들이 계산되고 제2클럭(CLK_B)가 '논리 하이'일 때 제2경로(path2)를 선택하여 d_i값들이 계산된다.

본 발명의 기술 사상은 상기 바람직한 실시예에 따라 구체적으로 기술되었으나 상기한 실시예는 그 설명을 위한 것이며 그 제한을 위한 것이 아님을 주의하여야 한다. 또한, 본 발명의 기술 분야의 통상의 전문가라면 본 발명의 기술 사상의 범위내에서 다양한 실시예가 가능함을 이해할 수 있을 것이다.

발명의 효과

상기와 같이 본 발명은 S-Box를 한번에 하나씩만을 사용함으로써 S-Box 치환부가 차지하는 면적을 반으로 감소시킨다. 그래서, 효율적으로 회로들을 배치할 수 있고, 적은 칩 면적으로 넷 다이(Net Die) 수를 증대시키므로 코스트(Cost) 향상을 가져온다.

(57) 청구의 범위

청구항 1.

데이터 초기치환부, 제1 클럭, 제2 클럭 및 제3 클럭을 이용하는 n단 (n은 짝수) 파이프라인 구조를 갖는 데이터 변형부 및 데이터 역초기치환부를 포함하고, 데이터 암호화 표준 알고리즘을 사용하여 평문 텍스트 블록의 암호화를 수행하기 위한 암호화 장치에 있어서,

n/3 개의 48비트의 데이터 중에 하나를 선택하기 위한 멀티플렉서;

상기 멀티플렉서에서 출력된 48비트 중에서 6비트의 어드레스 8개를 각각 입력받아 4비트의 데이터 8 개를 각각 출력하는 8 개의 S-Box를 포함하는 S-Box 치환부;

상기 32비트의 데이터를 n/3개의 출력으로 분배하는 디멀티플렉서; 및

제4클럭 및 제5클럭에 따라 상기 멀티플렉서 및 상기 디멀티플렉서를 제어하는 제어기를 포함하고,

상기 제4 클럭 및 상기 제5 클럭은 상기 제1 클럭, 상기 제2 클럭 및 상기 제3 클럭보다 1/3 주기만큼 빠르며, 상기 제4클럭과 제5클럭은 서로 반전된 신호임을 특징으로 하는 암호화 장치.

청구항 2.

삭제

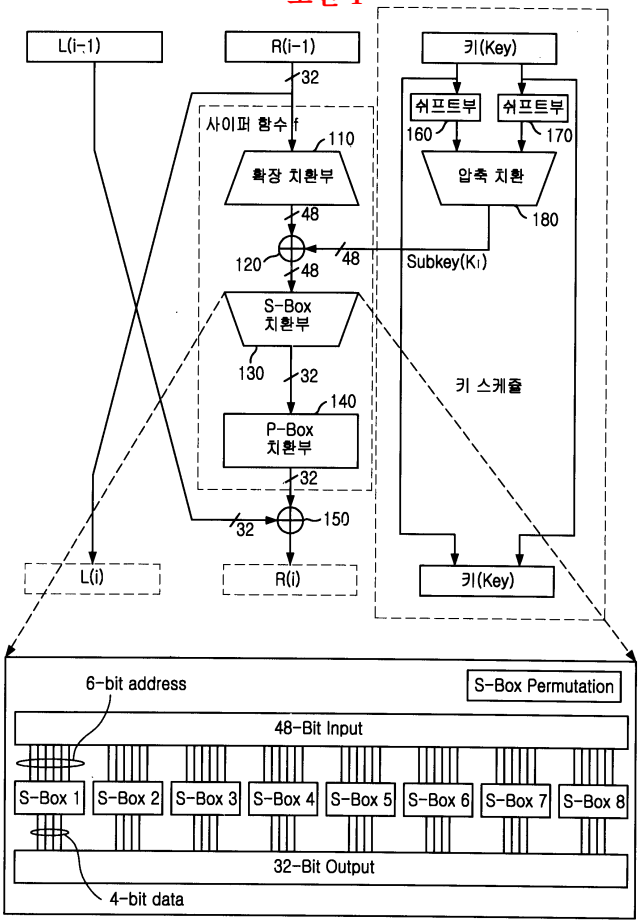
청구항 3.

제 1 항에 있어서,

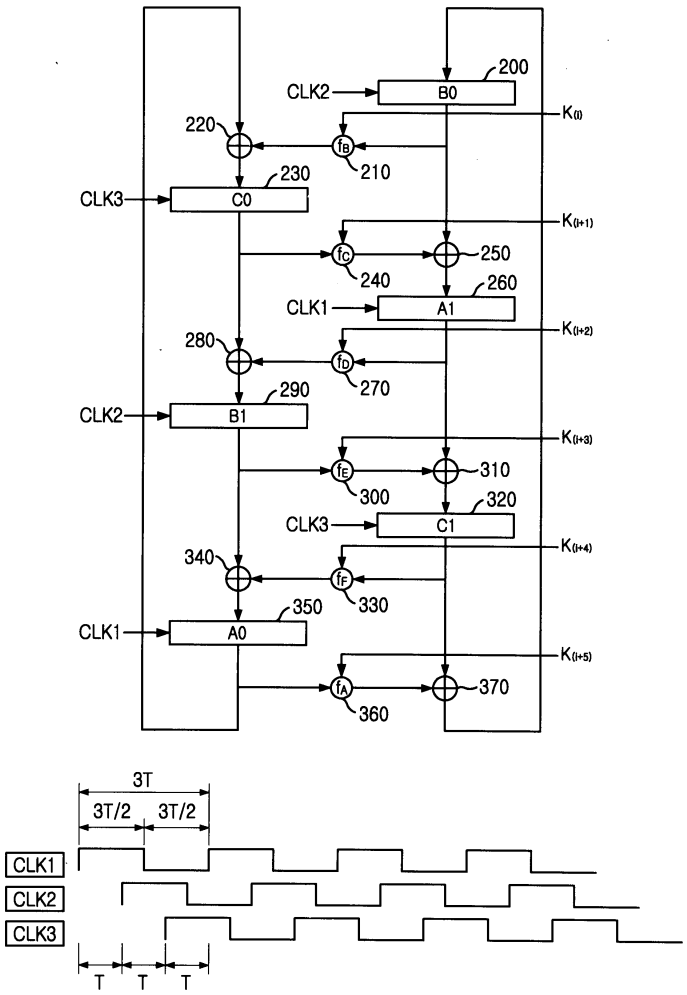
상기 멀티플렉서와 디멀티플렉서는 물리적으로 존재하는 n/3 개의 입력과 출력 경로를 시분할하여 데이터의 충돌을 방지하는 것을 특징으로 하는 암호화 장치.

도면

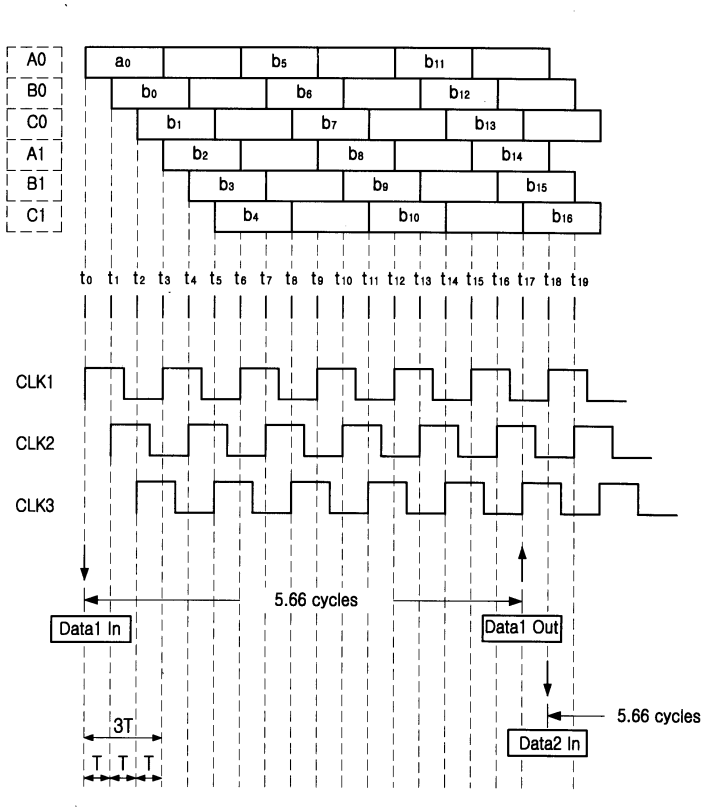
도면 1


L(i-1)
R(i-1)
32
사이퍼 함수 f
110
확장 치환부
48
120
48
48
S-Box
치환부
130
32
140
P-Box
치환부
32
32
150
L(i)
R(i)
키(Key)
쉬프트부
쉬프트부
160
170
압축 치환
180
Subkey(Ki)
키 스케줄
키(Key)
6-bit address
S-Box Permutation
48-Bit Input
S-Box 1
S-Box 2
S-Box 3
S-Box 4
S-Box 5
S-Box 6
S-Box 7
S-Box 8
32-Bit Output
4-bit data

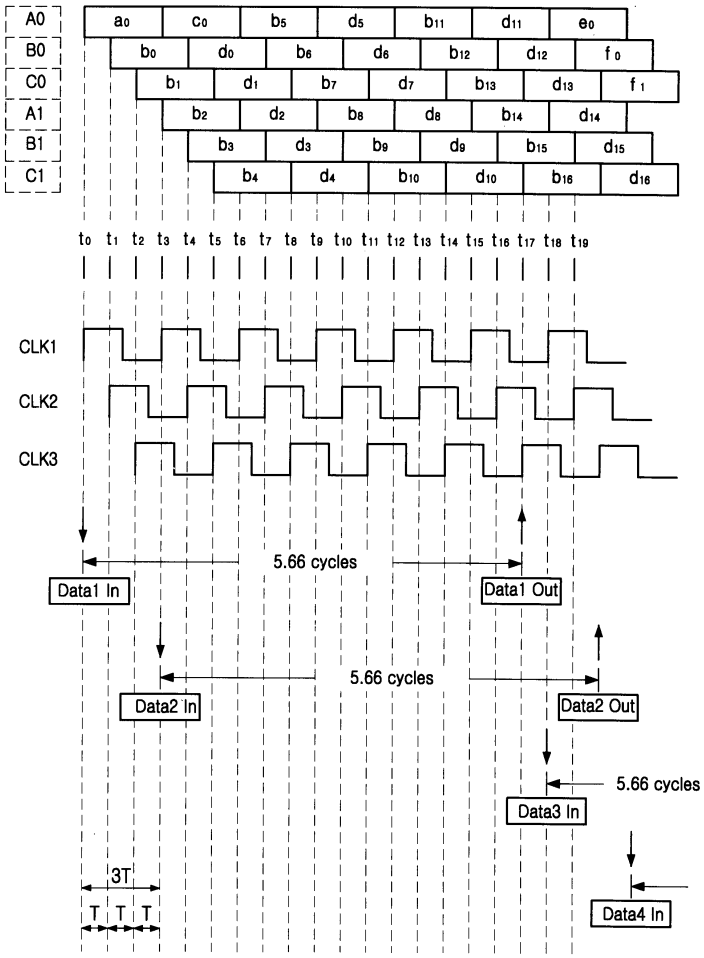
도면 2


CLK2
B0
200
K(i)
220
fB
210
CLK3
C0
230
K(i+1)
fC
250
240
CLK1
A1
260
K(i+2)
280
fD
270
CLK2
B1
290
K(i+3)
fE
310
300
CLK3
C1
320
K(i+4)
340
fF
330
CLK1
A0
350
K(i+5)
fA
370
360
3T
3T/2
3T/2
CLK1
CLK2
CLK3
T
T
T

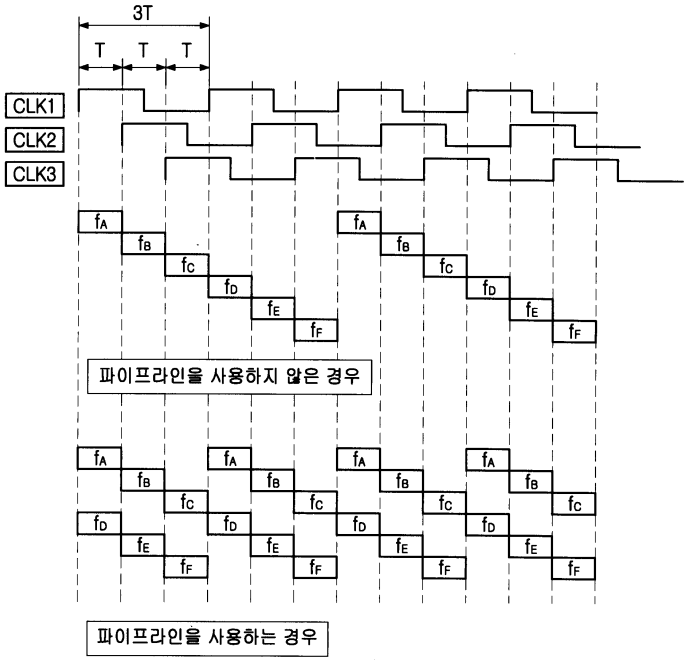
도면 3


A0
B0
C0
A1
B1
C1
a0
b5
b11
b0
b6
b12
b1
b7
b13
b2
b8
b14
b3
b9
b15
b4
b10
b16
t0 t1 t2 t3 t4 t5 t6 t7 t8 t9 t10 t11 t12 t13 t14 t15 t16 t17 t18 t19
CLK1
CLK2
CLK3
5.66 cycles
Data1 In
Data1 Out
5.66 cycles
Data2 In
3T
T T T

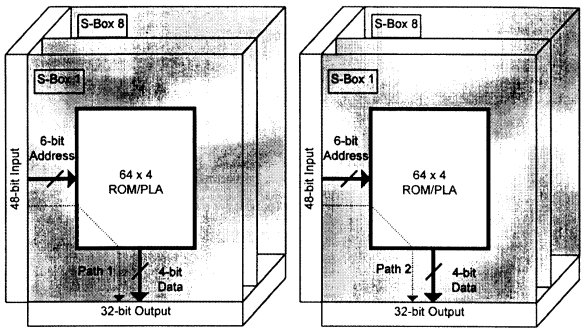
도면 4


A0
B0
C0
A1
B1
C1
a0 c0 b5 d5 b11 d11 e0
b0 d0 b6 d6 b12 d12 f0
b1 d1 b7 d7 b13 d13 f1
b2 d2 b8 d8 b14 d14
b3 d3 b9 d9 b15 d15
b4 d4 b10 d10 b16 d16
t0 t1 t2 t3 t4 t5 t6 t7 t8 t9 t10 t11 t12 t13 t14 t15 t16 t17 t18 t19
CLK1
CLK2
CLK3
5.66 cycles
Data1 In
Data1 Out
5.66 cycles
Data2 In
Data2 Out
5.66 cycles
Data3 In
Data4 In
3T
T T T

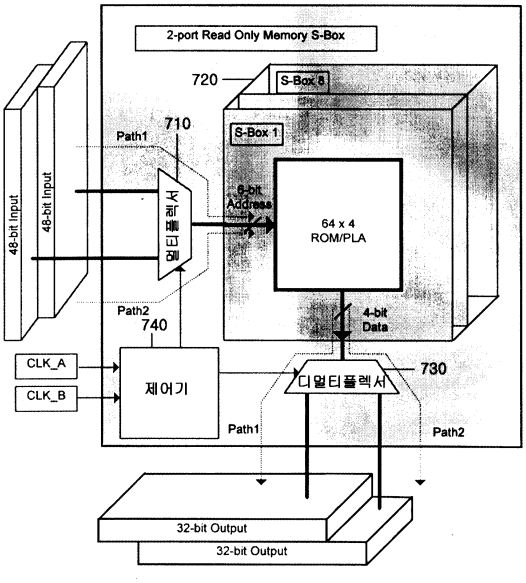
도면 5


3T
T
T
T
CLK1
CLK2
CLK3
fA
fB
fC
fD
fE
fF
파이프라인을 사용하지 않은 경우
파이프라인을 사용하는 경우

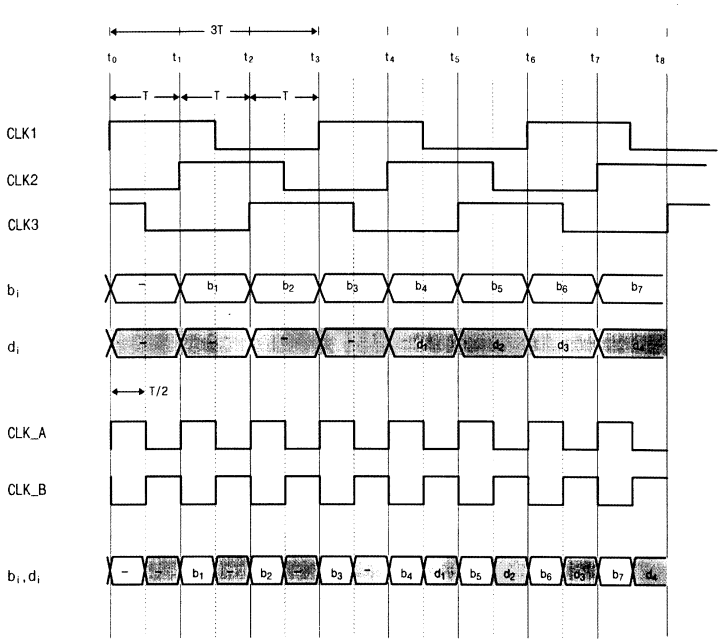
도면 6


S-Box 8
S-Box 1
6-bit Address
64 x 4 ROM/PLA
48-bit Input
Path 1
Path 2
4-bit Data
32-bit Output

도면 7


2-port Read Only Memory S-Box
720
S-Box 8
S-Box 1
710
Path1
48-bit Input
48-bit Input
멀티플렉서
6-bit Address
64 x 4 ROM/PLA
Path2
740
4-bit Data
CLK_A
CLK_B
제어기
730
디멀티플렉서
Path1
Path2
32-bit Output
32-bit Output

도면 8



–	b_1	b_2	b_3	b_4	b_5	b_6	b_7
–	$a_0 \oplus f(b_0, K_1)$	$b_0 \oplus f(b_1, K_2)$	$b_1 \oplus f(b_2, K_3)$	$b_2 \oplus f(b_3, K_4)$	$b_3 \oplus f(b_4, K_5)$	$b_4 \oplus f(b_5, K_6)$	$b_5 \oplus f(b_6, K_7)$

–	–	–	–	d_1	d_2	d_3	d_4
–	–	–	–	$c_0 \oplus f(d_0, K_1)$	$d_0 \oplus f(d_1, K_2)$	$d_1 \oplus f(d_2, K_3)$	$d_2 \oplus f(d_3, K_4)$