Data Encryption Standard (DES)

https://youtu.be/1SGoGddZnd0

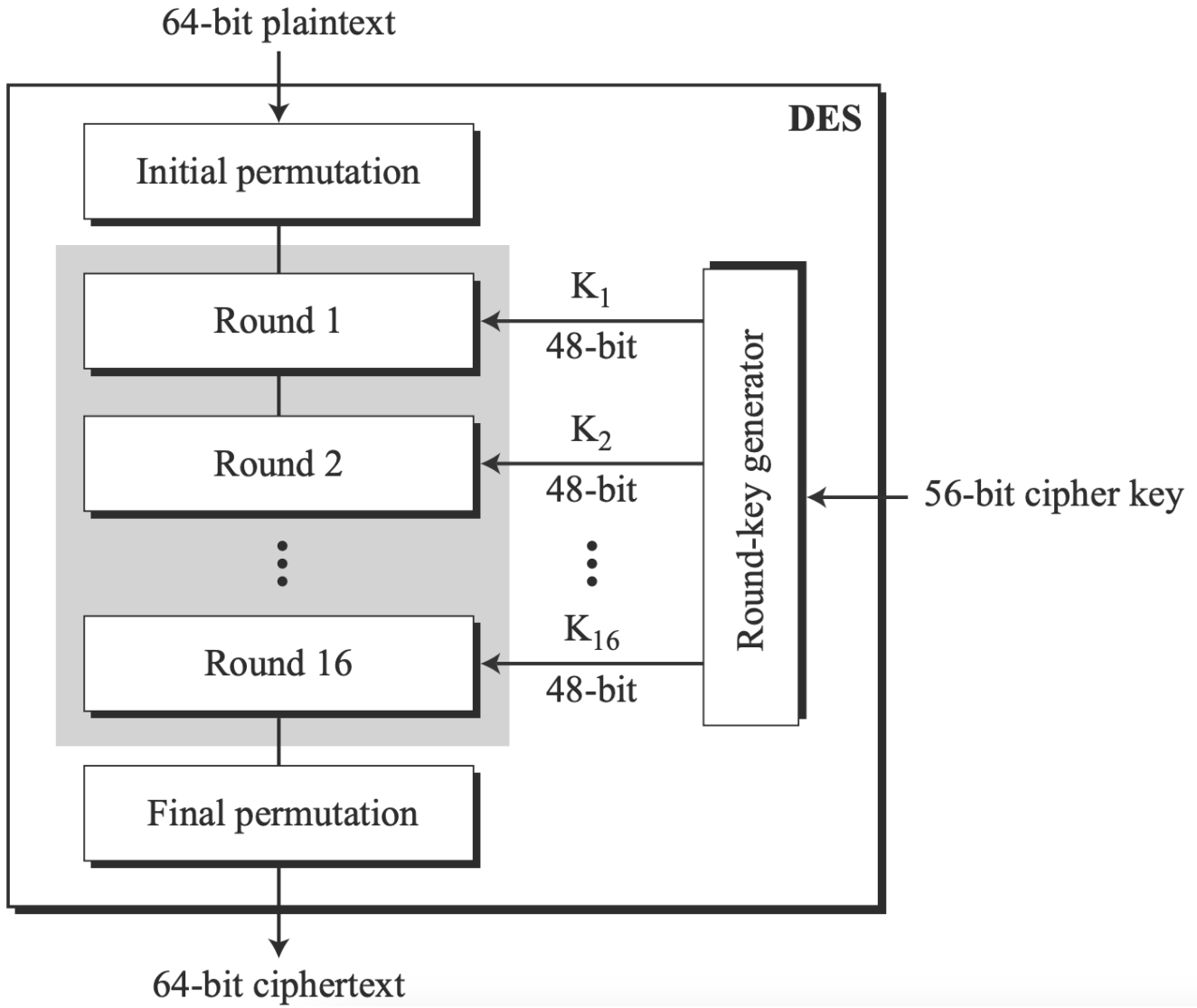
송경주

Data Encryption Standard (DES)

- Plaintext : 64bit
- Ciphertext : 64bit
- Master key : 56 bit
- Round key : 48 bit
- 라운드 수 : 16

<DES 암호화 과정>

1. Key generator
2. Initial permutation
3. Round (DES function)
4. Final permutation



Data Encryption Standard (DES)

- Initial Permutation

- 64bit 입력에 대한 정렬을 바꿈, 이때 정렬은 미리 정해진 규칙을 따름 (permutation table)
- 두 permutation이 암호 강도에 영향을 미치지는 않음

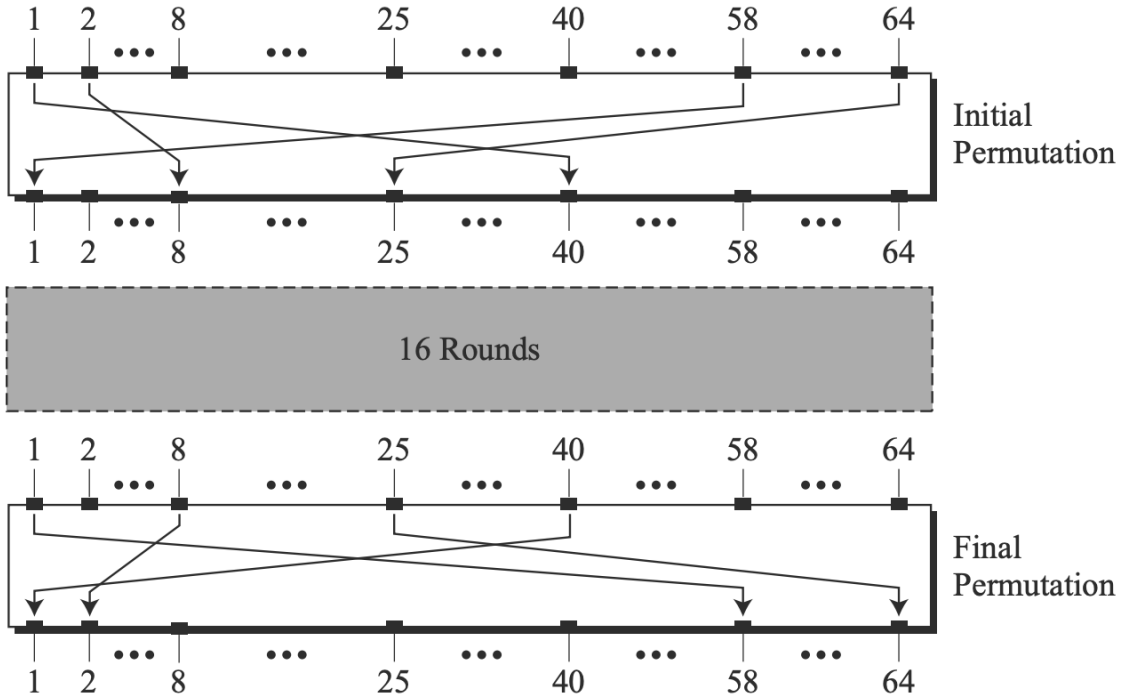


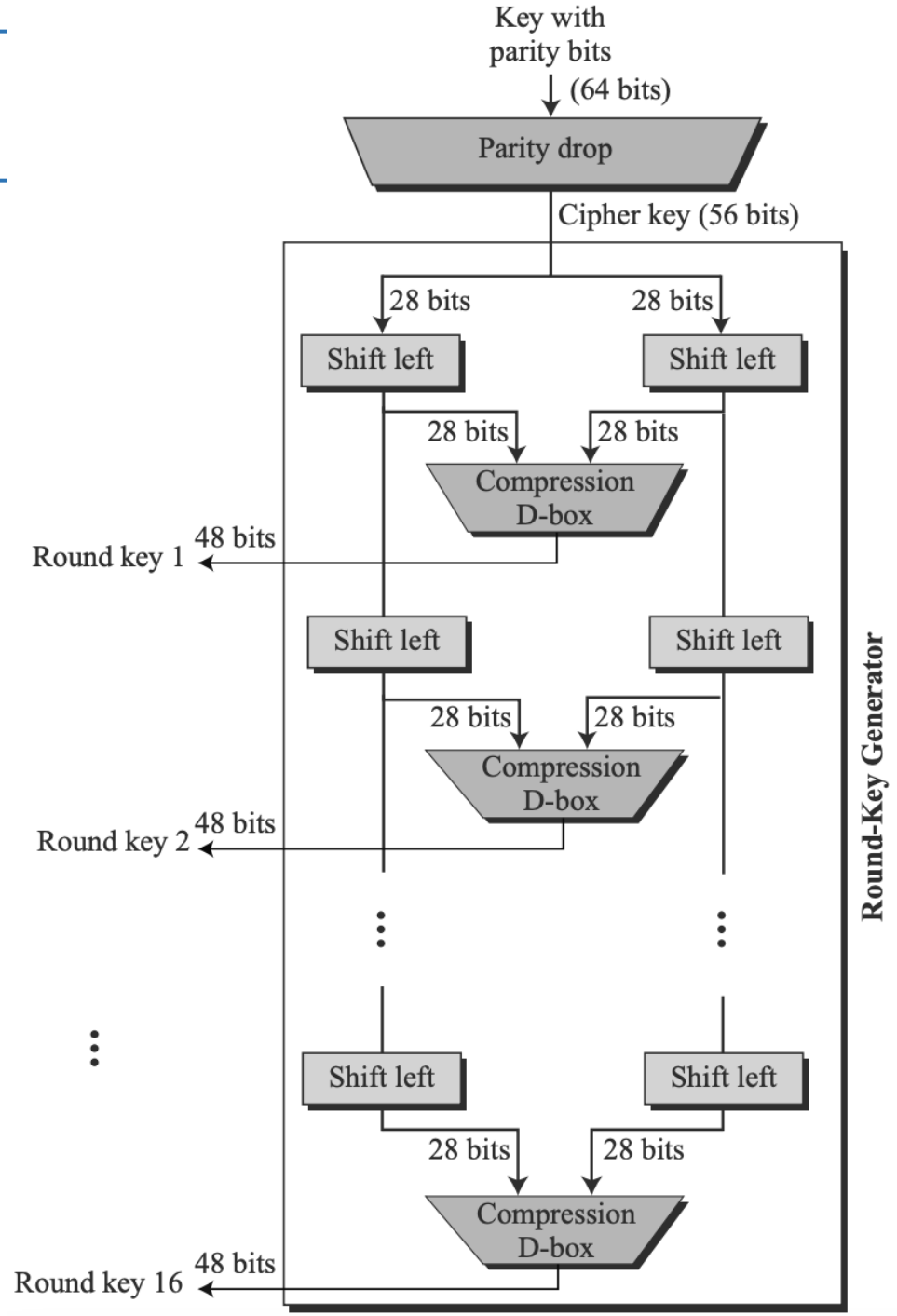
Table 6.1 *Initial and final permutation tables*

Initial Permutation	*Final Permutation*
58 50 42 34 26 18 10 02	40 08 48 16 56 24 64 32
60 52 44 36 28 20 12 04	39 07 47 15 55 23 63 31
62 54 46 38 30 22 14 06	38 06 46 14 54 22 62 30
64 56 48 40 32 24 16 08	37 05 45 13 53 21 61 29
57 49 41 33 25 17 09 01	36 04 44 12 52 20 60 28
59 51 43 35 27 19 11 03	35 03 43 11 51 19 59 27
61 53 45 37 29 21 13 05	34 02 42 10 50 18 58 26
63 55 47 39 31 23 15 07	33 01 41 09 49 17 57 25

Data Encryption Standard (DES)

- Key generator

- 64bit master 키를 사용하여 라운드키 생성
- 맨 처음 parity drop 함수를 통해 56bit key만 사용
- 56bit key를 28||28 로 나누어 Shift left 진행
- Compression D-box를 사용하여 56bit인풋을 사용하여 48bit 라운드 키 생성
- 라운드 키 크기 : 48bit



Shifting

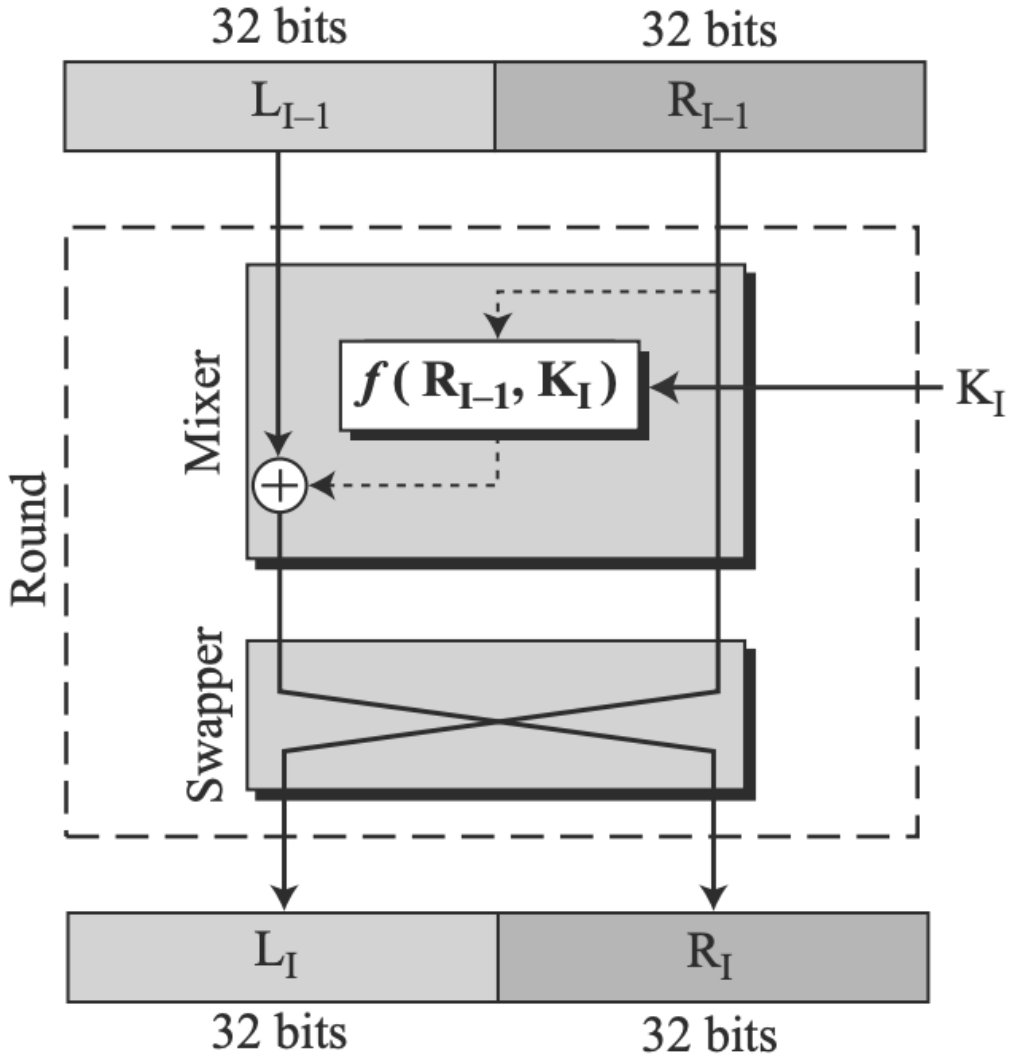
Rounds	Shift
1, 2, 9, 16	one bit
Others	two bits

Data Encryption Standard (DES)

- Round function

- 64 bit의 input을 32bit 씩 나눠서 진행
- [Left(L) || Right(R)] → [32 bit || 32 bit]
- R : f (R, K) 진행, 기존의 값 유지
- L : f (R, K) 과 XOR 연산이 수행됨

- **R : 다음 라운드의 L**
- **f (R, K) ⊕ L : 다음 라운드의 R**

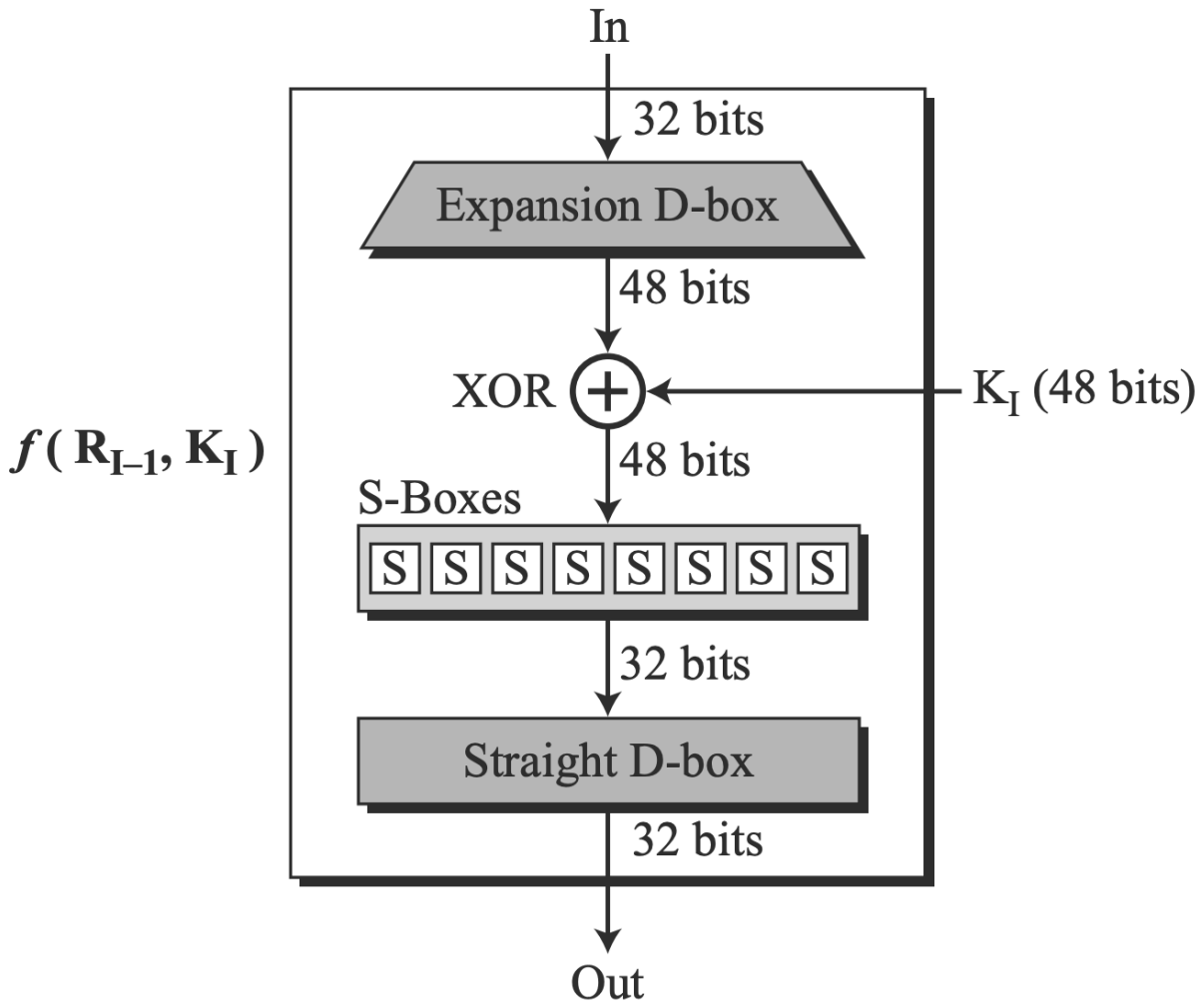


Data Encryption Standard (DES)

- Round function

- 크게 4가지 과정으로 진행됨
- 32bit 인풋에 대해 32bit 을 출력함

1. Expansion D-box
2. Round key XOR
3. S-Boxes
4. Straight D-box

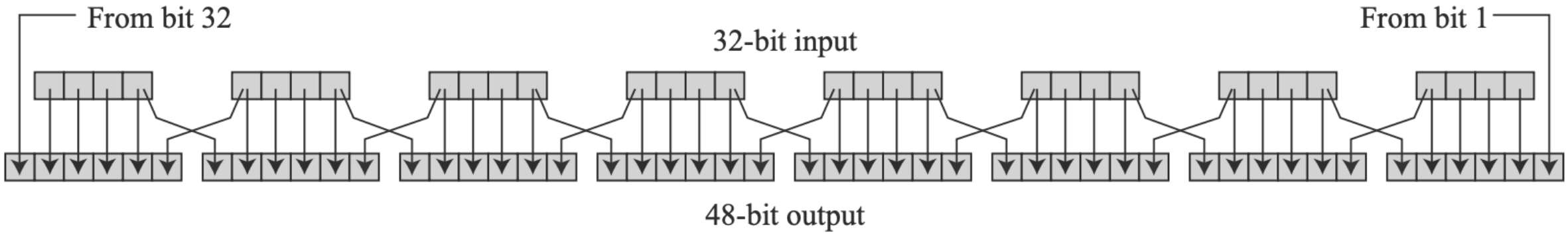


Data Encryption Standard (DES)

1. Expansion D-box

- 32bit 입력을 48bit 로 확장 시킴
- 32bit 를 4bit 씩 8개로 나누고 규칙에 따라 확장 진행

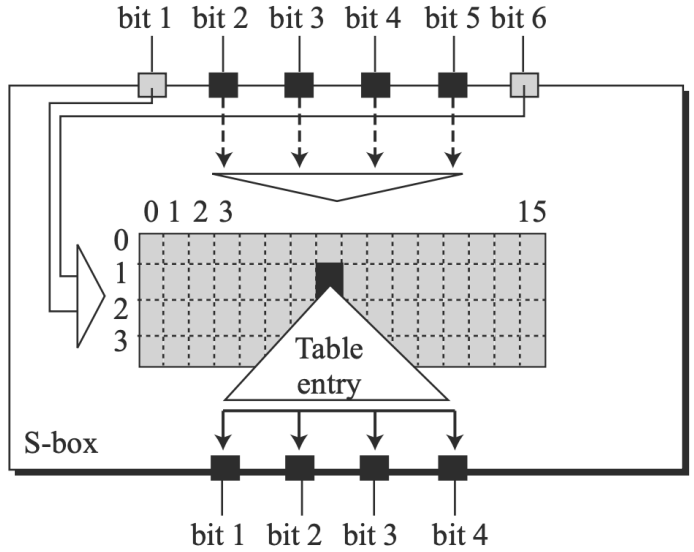
32	01	02	03	04	05
04	05	06	07	08	09
08	09	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	31	31	32	01

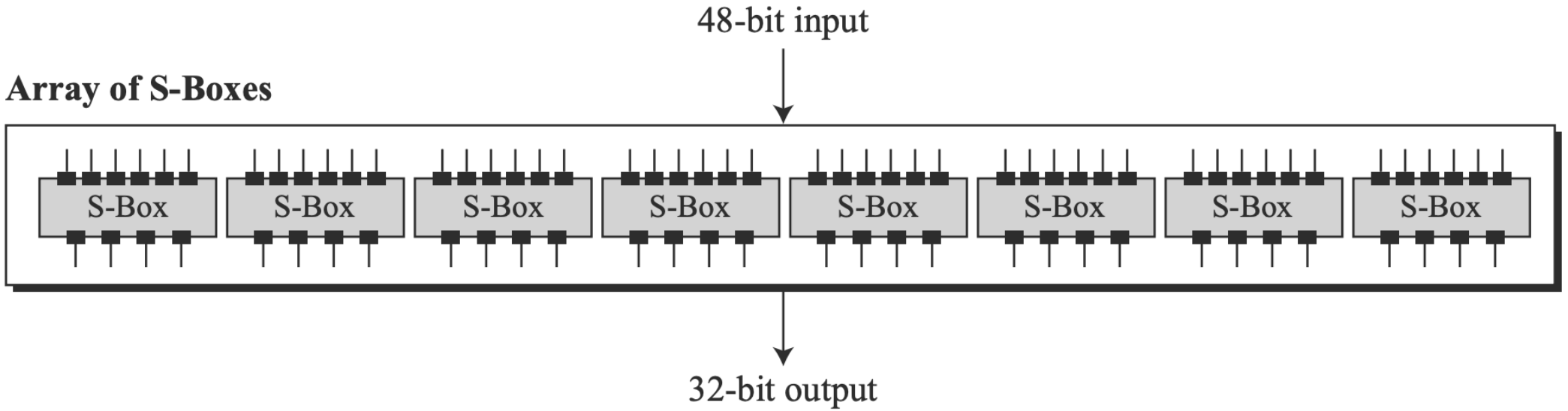


Data Encryption Standard (DES)

3. S-box

- DES의 S-box는 입력과 출력 길이가 다른 구조
- 입력 : 6 bit, 출력 : 4 bit
- DES function 입력 48bit 에 대해 6bit 씩 8개의 S-box 진행
- 8개의 S-box는 모두 다른 연산결과를 출력함 (독립적)





Data Encryption Standard (DES)

- 양자회로에서는 S-box equation 을 사용하여 구현

```
x26 = x6 ^ x25;
x27 = x1 & x8;
x28 = a2 | x27;
x29 = x26 ^ x28;
x30 = x1 | x8;
x31 = x30 ^ x6;
x32 = x5 & x14;
x33 = x32 ^ x8;
x34 = a2 & x33;
x35 = x31 ^ x34;
x36 = a5 | x35;
x37 = x29 ^ x36;
*out1 ^= x37;
```

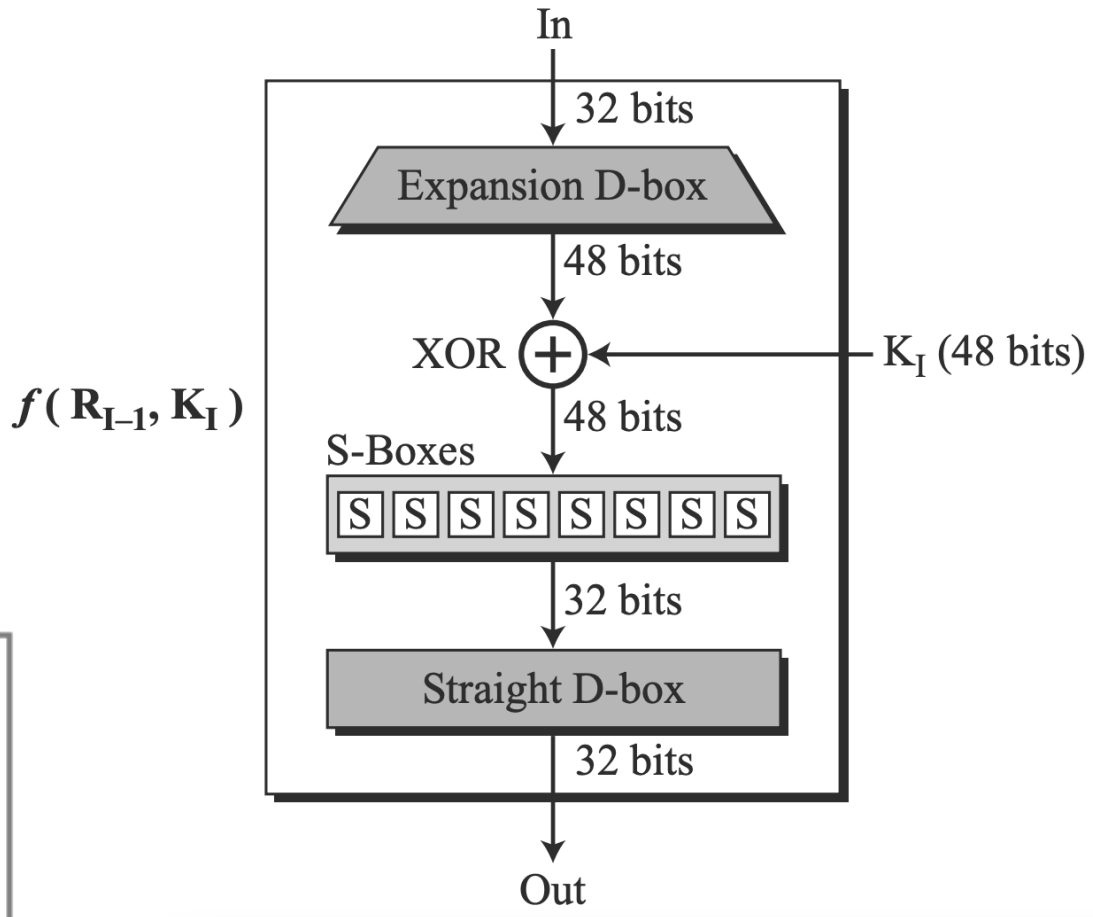
```
x1 = ~a4;
x2 = ~a1;
x3 = a4 ^ a3;
x4 = x3 ^ x2;
x5 = a3 | x2;
x6 = x5 & x1;
x7 = a6 | x6;
x8 = x4 ^ x7;
x9 = x1 | x2;
x10 = a6 & x9;
x11 = x7 ^ x10;
x12 = a2 | x11;
x13 = x8 ^ x12;
```

```
x11 = x7 ^ x10;
x12 = a2 | x11;
x13 = x8 ^ x12;
x14 = x9 ^ x13;
x15 = a6 | x14;
x16 = x1 ^ x15;
x17 = ~x14;
x18 = x17 & x3;
x19 = a2 | x18;
x20 = x16 ^ x19;
x21 = a5 | x20;
x22 = x13 ^ x21;
*out4 ^= x22;
```

Data Encryption Standard (DES)

4. Straight D-box

- 32bit의 S-box 결과에 대해 premutation 동작 수행
- Expansion D-box와 달리 인풋, 아웃풋 길이가 같음



16	07	20	21	29	12	28	17
01	15	23	26	05	18	31	10
02	08	24	14	32	27	03	09
19	13	30	06	22	11	04	25

Data Encryption Standard (DES)

- 양자회로 구현 결과

- S-box 구현에 많은 temp 큐비트가 사용됨
  → 아직 S-box 부분을 최적화 하지 않아 최적화 진행 시 큐비트 수 및 게이트 수를 줄일 수 있을 수도..
  → 큐비트 수를 많이 줄이는 방법도 가능할거 같지만 Depth랑 게이트 수가 너무 많이 늘어나서 비효율적이라고 예상됨

- 크게 최적화 할 부분이 없지만 key generator 부분에서 미미하게 큐비트 수(최대 8큐비트)를 줄일 수 있음

```
Gate counts:
    Allocate : 7288
    CCX : 3424
    CX : 11712
    Deallocate : 7288
    X : 7120
```

Q & A