[[IP core] Lõi IP mã hóa giải mã DES](https://nguyenquanicd.blogspot.com/2017/08/ip-core-loi-ip-ma-hoa-giai-ma-des.html)

Tác giả [Nguyễn Quân](https://www.blogger.com/profile/11739939602738380602) at 21:23  [DES](https://nguyenquanicd.blogspot.com/search/label/DES?&max-results=5), [IP core](https://nguyenquanicd.blogspot.com/search/label/IP%20core?&max-results=5), [Synchronous Design](https://nguyenquanicd.blogspot.com/search/label/Synchronous%20Design?&max-results=5), [System Verilog](https://nguyenquanicd.blogspot.com/search/label/System%20Verilog?&max-results=5), [Verilog](https://nguyenquanicd.blogspot.com/search/label/Verilog?&max-results=5)  [8 bình luận](https://nguyenquanicd.blogspot.com/2017/08/ip-core-loi-ip-ma-hoa-giai-ma-des.html#comment-form)

*Bài viết phân tích và mô tả một thiết kế lõi IP thực hiện mã hóa và giải mã DES. Đầu tiên, chi tiết về bộ mã hóa sẽ được phân tích. Sau đó, những điểm khác biệt của bộ giải mã sẽ được phân tích để tích hợp bộ mã hóa để tạo ra một thiết kế đầy đủ chức năng mã hóa và giải mã DES. Thiết kế có thể được hiện thực trên FPGA hoặc ASIC.*

Tham khảo bài: <http://nguyenquanicd.blogspot.com/2017/08/background-thuat-toan-ma-hoa-va-giai-ma.html>

Để hiểu rõ, các bạn hãy đọc từng bước được mô tả trong bài trên và so sánh với từng bước được phân tích trong bài này.

**1. Lõi mã hóa DES**

**2.1 Sơ đồ tín hiệu**

|  |
| --- |
| [A diagram of a computer code  Description automatically generated](https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEjVvJI3EcI4cD1CJAMpYaBtuj13uivsMpTXBRXX7tual2ZFqIiu2o6k4WFsVtepQTqG5_J5JLR2DQAcAU-FVdceeKR-ULIY632igGdfJ4Tgyru_sVz3VVRp3Pl5t-V2KyX8RySvOQi3FdT1/s1600/des_encipher.png) |
| Hình 1. Tín hiệu giao tiếp của bộ mã hóa DES |

|  |
| --- |
| [A white sheet with black text  Description automatically generated](https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEji0LyQdcUXbeizZYqOE4TTeI17Su9Z5TnTOUfVHlM_S3TR4F-3VaWG3NIZRsJwZ0oc3PMBsJY9Uq7kRrp2EZFGc0He8ssLiRfGZ-H18HpXLwGx2kFqudnt11jmCV1SpNtiixJVRc_h-CFl/s1600/des_encipher_1.png) |
| Hình 2. Mô tả các tín hiệu giao tiếp của lõi mã hóa DES |

*Chú ý: Trong chuẩn DES, việc đánh số thứ tự của chuỗi bit được thực hiện từ trái qua phải. Một tín hiệu được khai báo theo thứ tự từ MSB đến LSB nên trong phần thiết kế, giá trị các bảng hoán vị được thay đổi lại theo thứ tự bit này.Ví dụ: một tín hiệu được khai báo [63:0] thì bit 1 trong chuẩn DES ứng với bit 63, bit 2 ứng trong chuẩn mã hóa ứng với bit 62, ... và bit 64 trong chuẩn DES ứng với bit 0.*

*Các bạn cần lưu ý kỹ điều này trước khi đọc tiếp những phần mô tả dưới đây.*

**2.2 Mạch tính toán khóa vòng - KS**

Mạch sử dụng 2 thanh ghi để lưu lại giá trị Cn và Dn sau mỗi lần tính toán.

**2.2.1 Tính hoán vị PC-1**

Hoán vị thực chất là thay đổi trật tự sắp xếp các bit theo quy định của chuẩn. Như đã lưu ý, trong hoán vị PC-1 sau đây, tại bảng tạo ra c0, vị trí ô MSB là bit des\_key\_in[7] ứng với bit thứ 57 theo cách đánh số từ trái sang phải của chuẩn. Tương tự, ở bảng tạo ra d0, vị trí MSB là des\_key\_in[1] ứng với bit thứ 63 theo cách đánh số từ trái sang phải của chuẩn.

|  |
| --- |
| [A diagram of a number and number  Description automatically generated with medium confidence](https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEgQCmHL8ixuV3QChu6kX0xvzSmHdK4d0ieOH6tRsvOIji55SEaka5QlAFBp_gfdXFtSdJXKjYwcKYWeT6D0XcwwbZQkdM2mQQth80_Eg1IFt5SKKUbGcN9pRXM9KHefEW7muJzths5lCeQ5/s1600/des_encipher_pc1.png) |
| Hình 3. Tính hoán vị PC-1 - chỉ số trong từng ô vuông là vị trí bit của tín hiệu des\_key\_in[63:0] |

Ví dụ về đoạn code thực hiện hoán vị tạo ra c0:

**assign** c0 = {des\_key\_in[7], des\_key\_in[15], des\_key\_in[23], des\_key\_in[31], des\_key\_in[39], des\_key\_in[47], des\_key\_in[55],  
 des\_key\_in[63], des\_key\_in[6], des\_key\_in[14], des\_key\_in[22], des\_key\_in[30], des\_key\_in[38], des\_key\_in[46],  
 des\_key\_in[54], des\_key\_in[62], des\_key\_in[5], des\_key\_in[13], des\_key\_in[21], des\_key\_in[29], des\_key\_in[37],  
 des\_key\_in[45], des\_key\_in[53], des\_key\_in[61], des\_key\_in[4], des\_key\_in[12], des\_key\_in[20], des\_key\_in[28]};

Đoạn code trên rất đơn giản là thực hiện sắp xếp lại các bit.

**2.2.2 Phép dịch trái**

Mạch này sử dụng 2 thanh ghi để lưu giá trị thứ n của C và D. Mạch này lấy đầu vào là:

* Giá trị C0 và D0 từ phép tính hoán vị PC-1
* Giá trị Cn-1 và Dn-1 từ 2 thanh ghi

Mạch sử dụng các tín hiệu điều khiển sau:

* rkey\_sel để lựa chọn khi nào sử dụng giá trị C0/D0 và khi nào sử dụng giá trị từ thanh ghi
* shift\_1 để xác định thời điểm dịch 1 bit và thời điểm dịch 2 bit
* key\_process để xác định thời điểm cần tính khóa vòng mới

Các tín hiệu trên sẽ được tạo ra từ các mạch khác và được mô tả trong các phần tiếp theo của bài này.

|  |
| --- |
| [A diagram of a computer program  Description automatically generated](https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEgjlr_LUXS_9TM1_lJN7f-vCyw6HGU5qO_Oc2kMntGD0_gXtkZH9DIBpsQJ6M0fQ0FG18jNnMeaItuIk0rK1Pt3dzKiPD6L24crsh_JpAJ_1oe0CeOhxfuY_LXfT7tlrW31ZU3eWKKYTV9S/s1600/des_encipher_shift.png) |
| Hình 4. Mạch dịch trái |

**2.2.3 Tính hoán vị PC-2**

Ngõ ra mạch dịch trái trên đây được hoán vị theo bảng PC\_2 như sau để tạo ra khóa vòng.

|  |
| --- |
| [A diagram of a number on a table  Description automatically generated](https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEiB81H9xVgtFR2KHQ4HiXGz_fX_uXkHPY_H1g0ycW0Qz9LIH4y3SsWDui-2eUBaYNR0OReqHK5UEos_U8sT1QmZCvcZQaHB-xUmfyih9ogyNtBLMmUN1gXerebj4NWjU5FEPQFqk_f8Dvub/s1600/des_encipher_pc2.png) |
| Hình 5. Mạch hoán vị PC-2 |

**2.3 Mạch tính hàm mã hóa f(R,K)**

**2.3.1 Tính hoán vị E**

Hoán vị E sẽ tạo ra giá trị đầu vào cho các bảng S.

|  |
| --- |
| [A screenshot of a computer  Description automatically generated](https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEgiVeFb-PXISFvc-85cmxRGDgMaiynIq7x9ptSYDeSM97iQ-gdMMp9mPI2kRjzKptoE39Mc7phhjqibRSg6pHRKzJHv19wrQbVh-WE9CijviekQLXoQD3qqLMpsiTng27J2sNfvhGJ1UFzt/s1600/des_encipher_E.png) |
| Hình 6. Hoán vị E và XOR với khóa vòng, kết quả được chia thành 8 nhóm 6 bit để dùng cho 8 bảng S |

**2.3.2 Thực thi tra bảng S**

Tra bảng S thực chất là thay thế giá trị đầu vào bằng một giá trị khác được quy định trong bảng.

|  |
| --- |
| [A table of equations with numbers  Description automatically generated](https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEipR1U6eAZ0CFc11dJB-W_-0xcTAo_L9DSVdil6oTZ0Xl-4f9RmjHRNsQY8B0U5M1zvRt_hvJfrF4vHNBKENhpasEBBrrmI-02adv_5-nGKW4WVtB4aWkAet1Aiyl3U2GctMCIC7VfaNdIC/s1600/des_encipher_S.png) |
| Hình 6. Thực thi tra bảng S |

Một đoạn code RTL thực thi tra bảng S1 sẽ được thể hiện dưới đây. Trong đó, mỗi phát biểu case(s1\_in[4:1]) tương ứng với việc tra 1 hàng của bảng S1:

**always**@ (\*) **begin**  
 **case** ({s1\_in[5], s1\_in[0]})  
 2'b00: **begin**  
 **case**(s1\_in[4:1])  
 4'd0: s1\_out = 4'd14;  
 4'd1: s1\_out = 4'd4;  
 4'd2: s1\_out = 4'd13;  
 4'd3: s1\_out = 4'd1;  
 4'd4: s1\_out = 4'd2;  
 4'd5: s1\_out = 4'd15;  
 4'd6: s1\_out = 4'd11;  
 4'd7: s1\_out = 4'd8;  
 4'd8: s1\_out = 4'd3;  
 4'd9: s1\_out = 4'd10;  
 4'd10: s1\_out = 4'd6;  
 4'd11: s1\_out = 4'd12;  
 4'd12: s1\_out = 4'd5;  
 4'd13: s1\_out = 4'd9;  
 4'd14: s1\_out = 4'd0;  
 **default**: s1\_out = 4'd7;  
 **endcase**  
 **end**  
 2'b01: **begin**  
 **case**(s1\_in[4:1])  
 4'd0: s1\_out = 4'd0;  
 4'd1: s1\_out = 4'd15;  
 4'd2: s1\_out = 4'd7;  
 4'd3: s1\_out = 4'd4;  
 4'd4: s1\_out = 4'd14;  
 4'd5: s1\_out = 4'd2;  
 4'd6: s1\_out = 4'd13;  
 4'd7: s1\_out = 4'd1;  
 4'd8: s1\_out = 4'd10;  
 4'd9: s1\_out = 4'd6;  
 4'd10: s1\_out = 4'd12;  
 4'd11: s1\_out = 4'd11;  
 4'd12: s1\_out = 4'd9;  
 4'd13: s1\_out = 4'd5;  
 4'd14: s1\_out = 4'd3;  
 **default**: s1\_out = 4'd8;  
 **endcase**  
 e**n**d  
 2'b10: **begin**  
 **case**(s1\_in[4:1])  
 4'd0: s1\_out = 4'd4;  
 4'd1: s1\_out = 4'd1;  
 4'd2: s1\_out = 4'd14;  
 4'd3: s1\_out = 4'd8;  
 4'd4: s1\_out = 4'd13;  
 4'd5: s1\_out = 4'd6;  
 4'd6: s1\_out = 4'd2;  
 4'd7: s1\_out = 4'd11;  
 4'd8: s1\_out = 4'd15;  
 4'd9: s1\_out = 4'd12;  
 4'd10: s1\_out = 4'd9;  
 4'd11: s1\_out = 4'd7;  
 4'd12: s1\_out = 4'd3;  
 4'd13: s1\_out = 4'd10;  
 4'd14: s1\_out = 4'd5;  
 **default**: s1\_out = 4'd0;  
 **endcase**  
 **end**  
 **default**: **begin**  
 case (s1\_in[4:1])  
 4'd0: s1\_out = 4'd15;  
 4'd1: s1\_out = 4'd12;  
 4'd2: s1\_out = 4'd8;  
 4'd3: s1\_out = 4'd2;  
 4'd4: s1\_out = 4'd4;  
 4'd5: s1\_out = 4'd9;  
 4'd6: s1\_out = 4'd1;  
 4'd7: s1\_out = 4'd7;  
 4'd8: s1\_out = 4'd5;  
 4'd9: s1\_out = 4'd11;  
 4'd10: s1\_out = 4'd3;  
 4'd11: s1\_out = 4'd14;  
 4'd12: s1\_out = 4'd10;  
 4'd13: s1\_out = 4'd0;  
 4'd14: s1\_out = 4'd6;  
 **default**: s1\_out = 4'd13;  
 **endcase**  
 **end**  
 **endcase**  
**end**

**2.3.3 Hoán vị P**

Sau khi tra các bảng S, giá trị sẽ được hoán vị theo bảng P để tạo ra giá trị hàm f(R,K).

|  |
| --- |
| [A diagram of a number  Description automatically generated](https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEixgLJr0yAhTNDPkHVxPDJ3phzS2hiQh98xKv6lCMTRKy44oXxsAaBJmnAkbZOI8OUWW65COWgP5xlg9eayt65lsGFwa5z-Xom_Igd_SpYdZsvgpZAm3cqVkYrB94DPjNsjYESh8HDhUYD4/s1600/des_encipher_P.png) |
| Hình 8. Hoán vị P để tạo giá trị hàm f(R,K) |

Đoạn RTL code thực hiện hoán vị P:

**assign** p\_in = {s1\_out, s2\_out, s3\_out, s4\_out, s5\_out, s6\_out, s7\_out, s8\_out};  
**assign** f\_value = {p\_in[16], p\_in[25], p\_in[12], p\_in[11],  
 p\_in[3], p\_in[20], p\_in[4], p\_in[15],  
 p\_in[31], p\_in[17], p\_in[9], p\_in[6],  
 p\_in[27], p\_in[14], p\_in[1], p\_in[22],  
 p\_in[30], p\_in[24], p\_in[8], p\_in[18],  
 p\_in[0], p\_in[5], p\_in[29], p\_in[23],  
 p\_in[13], p\_in[19], p\_in[2], p\_in[26],  
 p\_in[10], p\_in[21], p\_in[28], p\_in[7]};

**2.4 Mạch thực hiện thuật toán mã hóa DES**

**2.4.1 Hoán vị đầu vào - IP và lựa chọn giá trị cho mỗi vòng lặp mã hóa**

|  |
| --- |
| [A screenshot of a computer  Description automatically generated](https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEg97VUaUZc6NQ5mUW2SLkLU51qUMpZyHww-DlP08CHCe7oQ0aYScxWgS0gcXezyEudKzlnsQpU1_ju0eLZO8vLXA91VFgTDAeglqhJjJl1trxp0JLbC9R9Udln3HrN5knHzumagTCw7173k/s1600/des_encipher_IP.png) |
| Hình 9. Thực hiện hoán vị đầu vào và lựa chọn giá trị |

Hoán vị đầu vào tạo ra giá trị ngõ vào đầu tiên l0 và r0. Mạch sử dụng một tín hiệu điều khiển cipher\_sel để chọn giá trị tính toán là l0-r0 hoặc giá trị ln-rn (đây là gí trị lưu trong thanh ghi).

Đoạn RTL code thực hiện hoán vị đầu vào và lựa chọn giá trị cho mỗi vòng lặp mã hóa như sau:

**assign**ip\_input = {des\_data[6], des\_data[14], des\_data[22], des\_data[30], des\_data[38], des\_data[46], des\_data[54], des\_data[62],  
 des\_data[4], des\_data[12], des\_data[20], des\_data[28], des\_data[36], des\_data[44], des\_data[52], des\_data[60],  
 des\_data[2], des\_data[10], des\_data[18], des\_data[26], des\_data[34], des\_data[42], des\_data[50], des\_data[58],  
 des\_data[0], des\_data[8], des\_data[16], des\_data[24], des\_data[32], des\_data[40], des\_data[48], des\_data[56],  
 des\_data[7], des\_data[15], des\_data[23], des\_data[31], des\_data[39], des\_data[47], des\_data[55], des\_data[63],  
 des\_data[5], des\_data[13], des\_data[21], des\_data[29], des\_data[37], des\_data[45], des\_data[53], des\_data[61],  
 des\_data[3], des\_data[11], des\_data[19], des\_data[27], des\_data[35], des\_data[43], des\_data[51], des\_data[59],  
 des\_data[1], des\_data[9], des\_data[17], des\_data[25], des\_data[33], des\_data[41], des\_data[49], des\_data[57]};  
**assign** l0 = ip\_input[63:32];  
**assign** r0 = ip\_input[31:0];  
**assign** l\_input = (cipher\_sel)? l0: ln;  
**assign** r\_input = (cipher\_sel)? r0: rn;

**2.4.2 Lưu giá trị Ln và Rn, thực hiện hoán vị đảo IP-1**

|  |
| --- |
| [A diagram of a mathematical equation  Description automatically generated](https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEg-_cktHVB5zmygy1Ba1HQpi0i_KrlI1y3SQ3uyA7c9sKNFlbBZb98MsNJ2NZMr9Ru_r7nViMJ8QUlVtOhymh2IItS6pf_6SQE4hctcXn-c9jeXeiuhLo3hfohaUbnrgg8Wbg21PPg3pMWB/s1600/des_encipher_IP_1.png) |
| Hình 10. Tính, lưu giá trị ln-rn và thực hiện hoán vị đảo IP-1 |

Ngõ ra l\_input[31:0] từ mạch hoán vị và lựa chọn mục 2.4.1 được XOR với giá trị hàm f(R,K) trước khi lưu vào thanh ghi ln[31:0]. Giá trị 2 thanh ghi được ghép lại thành giá trị ngõ vào của hoán vị IP-1. Sau hoán vị IP-1, desc\_ciphertext[63:0] là giá trị của mỗi vòng lặp mã hóa và là giá trị mã hóa nếu bộ mã hóa đang ở vòng lặp cuối cùng.

RTL code của thanh ghi ln và rn như sau:

**always**@ (posedge clk) **begin**  
 **if**(cipher\_process) **begin**  
 ln <= `DELAY r\_input;  
 rn <= `DELAY l\_input ^ f\_value;  
 **end**  
**end**

RTL code của hoán vị đảo IP-1:

**assign** inv\_p\_input = {rn, ln};  
**assign** desc\_ciphertext = {inv\_p\_input[24], inv\_p\_input[56], inv\_p\_input[16], inv\_p\_input[48], inv\_p\_input[8], inv\_p\_input[40], inv\_p\_input[0], inv\_p\_input[32],  
 inv\_p\_input[25], inv\_p\_input[57], inv\_p\_input[17], inv\_p\_input[49], inv\_p\_input[9], inv\_p\_input[41], inv\_p\_input[1], inv\_p\_input[33],  
 inv\_p\_input[26], inv\_p\_input[58], inv\_p\_input[18], inv\_p\_input[50], inv\_p\_input[10], inv\_p\_input[42], inv\_p\_input[2], inv\_p\_input[34],  
 inv\_p\_input[27], inv\_p\_input[59], inv\_p\_input[19], inv\_p\_input[51], inv\_p\_input[11], inv\_p\_input[43], inv\_p\_input[3], inv\_p\_input[35],  
 inv\_p\_input[28], inv\_p\_input[60], inv\_p\_input[20], inv\_p\_input[52], inv\_p\_input[12], inv\_p\_input[44], inv\_p\_input[4], inv\_p\_input[36],  
 inv\_p\_input[29], inv\_p\_input[61], inv\_p\_input[21], inv\_p\_input[53], inv\_p\_input[13], inv\_p\_input[45], inv\_p\_input[5], inv\_p\_input[37],  
 inv\_p\_input[30], inv\_p\_input[62], inv\_p\_input[22], inv\_p\_input[54], inv\_p\_input[14], inv\_p\_input[46], inv\_p\_input[6], inv\_p\_input[38],  
 inv\_p\_input[31], inv\_p\_input[63], inv\_p\_input[23], inv\_p\_input[55], inv\_p\_input[15], inv\_p\_input[47], inv\_p\_input[7], inv\_p\_input[39]};

**2.5 Mạch giám sát quá trình mã hóa**

Phần mạch này tạo ra các tín hiệu điều khiển cho các mạch đã trình bày ở trên. Để giám sát quá trình mã hóa một bộ đếm 4 bit được sử dụng để xác định số lần lặp mã hóa. Đồng thời giá trị bộ đếm này được sử dụng để tạo thêm tín hiệu điều khiển khác. Ví dụ, tín hiệu shift\_1 tích cực tại 4 vị trí ứng với lần lặp thứ 1 (bộ đếm bằng 0), thứ 2 (Bộ đếm bằng 1), thứ 9 (bộ đếm bằng 8) và thứ 16 (Bộ đếm bằng 15).

|  |
| --- |
| [A diagram of a computer program  Description automatically generated](https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEgiWsKCrfop-n4LeH_Hm2xlIrQb9jIkFftczfz7wqbB4XgcgPmYBjlFscaHRtpaswkqVzzUk6REU2HCqrPYjDOw5sY4Pb92VynZt8p70OvgI86HwcpYR_yxf3BX4Tbntl0YsbBrU8ovV1ud/s1600/des_encipher_counter.png) |
| Hình 11. Bộ đếm giám sát quá trình mã hóa và tín hiệu xác định quá trình tính khóa vòng key\_process |

**3. Bộ giải mã DES**

Bộ giải mã được thực hiện dựa trên cấu trúc bộ mã hóa. Phần này chỉ nhấn mạnh các điểm được điều chỉnh để tích hợp phần giải mã vào mạch mã hóa để tạo ra một thiết kế có chức năng mã háo và giải mã hoàn chỉnh.

**3.1 Tín hiệu giao tiếp**

Tín hiệu giao tiếp được điều chỉnh như sau:

* Tách tín hiệu des\_cipher\_en thành 2 tín hiệu:
  + des\_encipher\_en: Khởi động quá trình mã hóa
  + des\_decipher\_en: Khởi động quá trình giải mã
* Tín hiệu des\_ciphertext[63:0] được chuyển thành des\_result[63:0] và tín hiệu này sẽ là giá trị bản mã của quá trình mã hóa hoặc bản rõ của quá trình giải mã

|  |
| --- |
| [A diagram of a computer code  Description automatically generated](https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEhe7tGwvW8WlwkxnQoEqbEM1SjrKqyB5IktvGIv-qrnDJgFiMPvxUAwNf0lRj3ynARjsgijowWYy0Fapk_IOidNIzueluQHyghS5kC53MYjhCKKIayHRNG5yJiAgEY6u8pZ0DUZWKpXupF5/s1600/image.png) |
| Hình 12. Sơ đồ tín hiệu giao tiếp của lõi tích hợp chức năng giải mã |

**3.2 Mạch tính khóa vòng**

Mạch này điều chỉnh:

* Tín hiệu rkey\_sel ở mạch tạo cin và din
* Điều chỉnh mạch của thanh ghi cn và dn

|  |
| --- |
| [A screenshot of a computer program  Description automatically generated](https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEicd96GbD43e4RiytOtMQ7a2_-faDkett9h4UtIa8KOVAC4esuz4EY8W6bdDnnaih8h4HtANYP6sfRttGBV9hZv3m5ffegEPdmvBX9SIu4h_fknJOL7TvAo1M-yE4Kgx3st41sM6x5XiPcM/s1600/decipher_key.png) |
| Hình 13. Mạch tính khóa vòng |

**3.3 Mạch tính hàm mã hóa f(R,K)**

Không có sự thay đổi

**3.4 Mạch thực hiện thuật toán giải mã**

Điều chỉnh mạch tạo l\_input[31:0] và r\_input[31:0].

|  |
| --- |
| [A diagram of a computer program  Description automatically generated](https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEicH4lubYG457tCdnVO6t0zuD4BDgjSiCuhYvOXBFBP0cRXtYRvgtfY7_RTy5gLdWjdmBxfBvC-qHiz8nQhfWtjSC5UmmdrRPngmmVx1FFRqLrUtVWOU076i1m2Hpmmy2GeXc0sDZi443dK/s1600/image.png) |
| Hình 14. Mạch tạo ngõ vào cho bảng hoán vị IP-1 |

**3.5 Mạch tạo giá trị ngõ ra bộ mã hóa và giải mã DES**

Điều chỉnh thêm cổng OR ở mạch MUX tạo ngõ vào bảng hoán vị IP-1.

|  |
| --- |
| [A diagram of a mathematical algorithm  Description automatically generated with medium confidence](https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEhKn3-m5NS4_JtYQ5JyAG9lRRFIN1CuRjWwGm-VztRXIQcPEsg57ZELgBLSelWbBqmhHsJVfYywXpRxB_ApXXKaFR8oo5VKSksrym2xqRBkzPnefNIFgjhFUP6RmIfTvmsOyqxbRQOUxdda/s1600/image.png) |
| Hình 15. Mạch tạo ngõ ra bộ mã hóa và giải mã DES |

**3.6 Mạch giám sát quá trình giải mã và mã hóa**

Mạch này có nhiều thay đổi để tách quá trình mã hóa và giải mã riêng. bạn đọc có thể so sánh và tự tìm hiểu.

|  |
| --- |
| [A diagram of a machine  Description automatically generated](https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEgDFme7VRYVU3zMgXrMefI3oOYVRgY3ulBi32UamJqlnbR6gTKcNU_2rtJ2kW20QPWAGS7w4oQiG8vRw8ZkWiviGsO_Mnzj7KQgpm32XmWpbb8tWwlrJlBE15lE9caqRrFcuXBgWP6VmEqr/s1600/en_de_cipher.png) |
| Hình 16. Mạch giám sát quá trình mã hóa và giải mã |

Một điểm khác biệt là, khi quá trình giải mã được khởi động, quá trình tính khóa vòng sẽ chạy để tính ra khóa vòng K16. Sau đó, quá trình giải mã mới thực sự được khởi động.

**4. RTL code và testbench**

[DES\_core.zip](http://www.mediafire.com/file/lhcrand1mohn7fr/DES_core.zip)

pass (nếu có): nguyenquanicd