```
    1: module des(ck,keyin,k,datin,p,e,f);
    2:     input   ck, keyin, datin;
    3:     input   f;
    4:     input   [64:1]  p, k;
    5:     output  [64:1]  e;
    6:
    7:     reg [4:0] kst;
    8:     reg [48:1] KS01, KS02, KS03, KS04, KS05, KS06, KS07, KS08, KS09, KS10,
    9:                KS11, KS12, KS13, KS14, KS15, KS16;
   10:     reg [56:1] PC1, CD1;
   11:     reg [64:1] IP1, IP2, IP3, IP4, IP5,IP6, IP7, IP8, IP9, IP10, IP11, IP12, IP13, I
P14, IP15, IP16,IP17, LR, e;
   12:
   13:     reg [1:0] key_state;
   14:     reg [3:0] data_state;
   15:
   16:     always @(posedge ck) begin
   17:         key_state = key_state + 1;
   18:         data_state = data_state << 1;
   19:         if (keyin == 1) begin
   20:             key_state = 2'b01;
   21:         end else if (datin == 1) begin
   22:             data_state[0] = 1'b1;
   23:         end
   24:
   25:         case (key_state)
   26:             2'b01: begin
   27:                 PC1[56:1] = { k[4],k[12],k[20],k[28],k[5],k[13],k[21],
   28:                               k[29],k[37],k[45],k[53],k[61],k[6],k[14],
   29:                               k[22],k[30],k[38],k[46],k[54],k[62],k[7],
   30:                               k[15],k[23],k[31],k[39],k[47],k[55],k[63],
   31:
   32:                               k[36],k[44],k[52],k[60],k[3],k[11],k[19],
   33:                               k[27],k[35],k[43],k[51],k[59],k[2],k[10],
   34:                               k[18],k[26],k[34],k[42],k[50],k[58],k[1],
   35:                               k[9],k[17],k[25],k[33],k[41],k[49],k[57] };
   36:                 if (f == 1) begin
   37:                     PC1[56:1] = { PC1[29], PC1[56:30],PC1[1],PC1[28:2] };   KS01[48:
1] = PC2(PC1[56:1]);
   38:                     PC1[56:1] = { PC1[29], PC1[56:30],PC1[1],PC1[28:2] };   KS02[48:
1] = PC2(PC1[56:1]);
   39:                     PC1[56:1] = { PC1[30:29], PC1[56:31],PC1[2:1],PC1[28:3] };  KS03
[48:1] = PC2(PC1[56:1]);
   40:                     PC1[56:1] = { PC1[30:29], PC1[56:31],PC1[2:1],PC1[28:3] };  KS04
[48:1] = PC2(PC1[56:1]);
   41:                     PC1[56:1] = { PC1[30:29], PC1[56:31],PC1[2:1],PC1[28:3] };  KS05
[48:1] = PC2(PC1[56:1]);
   42:                     PC1[56:1] = { PC1[30:29], PC1[56:31],PC1[2:1],PC1[28:3] };  KS06
[48:1] = PC2(PC1[56:1]);
   43:                     PC1[56:1] = { PC1[30:29], PC1[56:31],PC1[2:1],PC1[28:3] };  KS07
[48:1] = PC2(PC1[56:1]);
   44:                     PC1[56:1] = { PC1[30:29], PC1[56:31],PC1[2:1],PC1[28:3] };  KS08
[48:1] = PC2(PC1[56:1]);
   45:                 end else begin
   46:                     PC1[56:1] = { PC1[29], PC1[56:30],PC1[1],PC1[28:2] };   KS16[48:
1] = PC2(PC1[56:1]);
   47:                     PC1[56:1] = { PC1[29], PC1[56:30],PC1[1],PC1[28:2] };   KS15[48:
1] = PC2(PC1[56:1]);
   48:                     PC1[56:1] = { PC1[30:29], PC1[56:31],PC1[2:1],PC1[28:3] };  KS14
[48:1] = PC2(PC1[56:1]);
   49:                     PC1[56:1] = { PC1[30:29], PC1[56:31],PC1[2:1],PC1[28:3] };  KS13
[48:1] = PC2(PC1[56:1]);
   50:                     PC1[56:1] = { PC1[30:29], PC1[56:31],PC1[2:1],PC1[28:3] };  KS12
[48:1] = PC2(PC1[56:1]);
   51:                     PC1[56:1] = { PC1[30:29], PC1[56:31],PC1[2:1],PC1[28:3] };  KS11
[48:1] = PC2(PC1[56:1]);
```

```
   52:                         PC1[56:1] = { PC1[30:29], PC1[56:31],PC1[2:1],PC1[28:3] };  KS10
[48:1] = PC2(PC1[56:1]);
   53:                         PC1[56:1] = { PC1[30:29], PC1[56:31],PC1[2:1],PC1[28:3] };  KS09
[48:1] = PC2(PC1[56:1]);
   54:                     end
   55:                 end
   56:                 2'b10: begin
   57:                     if (f == 1) begin
   58:                         PC1[56:1] = { PC1[29], PC1[56:30],PC1[1],PC1[28:2] };   KS09[48:
1] = PC2(PC1[56:1]);
   59:                         PC1[56:1] = { PC1[30:29], PC1[56:31],PC1[2:1],PC1[28:3] };  KS10
[48:1] = PC2(PC1[56:1]);
   60:                         PC1[56:1] = { PC1[30:29], PC1[56:31],PC1[2:1],PC1[28:3] };  KS11
[48:1] = PC2(PC1[56:1]);
   61:                         PC1[56:1] = { PC1[30:29], PC1[56:31],PC1[2:1],PC1[28:3] };  KS12
[48:1] = PC2(PC1[56:1]);
   62:                         PC1[56:1] = { PC1[30:29], PC1[56:31],PC1[2:1],PC1[28:3] };  KS13
[48:1] = PC2(PC1[56:1]);
   63:                         PC1[56:1] = { PC1[30:29], PC1[56:31],PC1[2:1],PC1[28:3] };  KS14
[48:1] = PC2(PC1[56:1]);
   64:                         PC1[56:1] = { PC1[30:29], PC1[56:31],PC1[2:1],PC1[28:3] };  KS15
[48:1] = PC2(PC1[56:1]);
   65:                         PC1[56:1] = { PC1[29], PC1[56:30],PC1[1],PC1[28:2] };   KS16[48:
1] = PC2(PC1[56:1]);
   66:                     end else begin
   67:                         PC1[56:1] = { PC1[29], PC1[56:30],PC1[1],PC1[28:2] };   KS08[48:
1] = PC2(PC1[56:1]);
   68:                         PC1[56:1] = { PC1[30:29], PC1[56:31],PC1[2:1],PC1[28:3] };  KS07
[48:1] = PC2(PC1[56:1]);
   69:                         PC1[56:1] = { PC1[30:29], PC1[56:31],PC1[2:1],PC1[28:3] };  KS06
[48:1] = PC2(PC1[56:1]);
   70:                         PC1[56:1] = { PC1[30:29], PC1[56:31],PC1[2:1],PC1[28:3] };  KS05
[48:1] = PC2(PC1[56:1]);
   71:                         PC1[56:1] = { PC1[30:29], PC1[56:31],PC1[2:1],PC1[28:3] };  KS04
[48:1] = PC2(PC1[56:1]);
   72:                         PC1[56:1] = { PC1[30:29], PC1[56:31],PC1[2:1],PC1[28:3] };  KS03
[48:1] = PC2(PC1[56:1]);
   73:                         PC1[56:1] = { PC1[30:29], PC1[56:31],PC1[2:1],PC1[28:3] };  KS02
[48:1] = PC2(PC1[56:1]);
   74:                         PC1[56:1] = { PC1[29], PC1[56:30],PC1[1],PC1[28:2] };   KS01[48:
1] = PC2(PC1[56:1]);
   75:                     end
   76:                 end
   77:             endcase
   78:
   79:             if (data_state[0]) begin
   80:                 IP1[64:1] <= { p[7],  p[15], p[23], p[31], p[39], p[47], p[55], p[63],
   81:                                p[5],  p[13], p[21], p[29], p[37], p[45], p[53], p[61],
   82:                                p[3],  p[11], p[19], p[27], p[35], p[43], p[51], p[59],
   83:                                p[1],  p[9],  p[17], p[25], p[33], p[41], p[49], p[57],
   84:                                p[8],  p[16], p[24], p[32], p[40], p[48], p[56], p[64],
   85:                                p[6],  p[14], p[22], p[30], p[38], p[46], p[54], p[62],
   86:                                p[4],  p[12], p[20], p[28], p[36], p[44], p[52], p[60],
   87:                                p[2],  p[10], p[18], p[26], p[34], p[42], p[50], p[58] };
   88:             end
   89:             if (data_state[1]) begin
   90:                 IP2 = des1( IP1, KS01 );
   91:                 IP3 = des1( IP2, KS02 );
   92:                 IP4 = des1( IP3, KS03 );
   93:                 IP5 = des1( IP4, KS04 );
   94:                 IP6 = des1( IP5, KS05 );
   95:                 IP7 = des1( IP6, KS06 );
   96:                 IP8 = des1( IP7, KS07 );
   97:                 IP9 <= des1( IP8, KS08 );
   98:             end
   99:             if (data_state[2]) begin
```

```
  100:             IP10 = des1( IP9, KS09 );
  101:             IP11 = des1( IP10, KS10 );
  102:             IP12 = des1( IP11, KS11 );
  103:             IP13 = des1( IP12, KS12 );
  104:             IP14 = des1( IP13, KS13 );
  105:             IP15 = des1( IP14, KS14 );
  106:             IP16 = des1( IP15, KS15 );
  107:             IP17 = des1( IP16, KS16 );
  108:             LR <= {IP17[32:1], IP17[64:33]};
  109:         end
  110:         if (data_state[3]) begin
  111:             // R=IP1[64:33], L=IP[32:1]
  112:             e <= { LR[25], LR[57], LR[17], LR[49], LR[9], LR[41], LR[1], LR[33],
  113:                    LR[26], LR[58], LR[18], LR[50], LR[10], LR[42], LR[2], LR[34],
  114:                    LR[27], LR[59], LR[19], LR[51], LR[11], LR[43], LR[3], LR[35],
  115:                    LR[28], LR[60], LR[20], LR[52], LR[12], LR[44], LR[4], LR[36],
  116:                    LR[29], LR[61], LR[21], LR[53], LR[13], LR[45], LR[5], LR[37],
  117:                    LR[30], LR[62], LR[22], LR[54], LR[14], LR[46], LR[6], LR[38],
  118:                    LR[31], LR[63], LR[23], LR[55], LR[15], LR[47], LR[7], LR[39],
  119:                    LR[32], LR[64], LR[24], LR[56], LR[16], LR[48], LR[8], LR[40] };
  120:         end
  121:     end
  122:
  123:     function [63:0] des1;
  124:         input [64:1] IP;
  125:         input [48:1] KS;
  126:         reg [48:1] R, RK;
  127:         reg [32:1] RS,P;
  128:
  129:         begin
  130:             R[48:1] = e2(IP[64:33]);
  131:             RK[48:1] = R^KS;
  132:             RS[4:1]=s1(RK[6:1]);
  133:             RS[8:5]=s2(RK[12:7]);
  134:             RS[12:9]=s3(RK[18:13]);
  135:             RS[16:13]=s4(RK[24:19]);
  136:             RS[20:17]=s5(RK[30:25]);
  137:             RS[24:21]=s6(RK[36:31]);
  138:             RS[28:25]=s7(RK[42:37]);
  139:             RS[32:29]=s8(RK[48:43]);
  140:             P[32:1] = { RS[25], RS[4], RS[11], RS[22], RS[6], RS[30], RS[13], RS[19]
,
  141:                         RS[9], RS[3], RS[27], RS[32], RS[14], RS[24], RS[8], RS[2],
  142:                         RS[10], RS[31], RS[18], RS[5], RS[26], RS[23], RS[15], RS[1]
,
  143:                         RS[17], RS[28], RS[12], RS[29], RS[21], RS[20], RS[7], RS[16
] };
  144:             des1 = { IP[32:1]^P , IP[64:33] };
  145:         end
  146:     endfunction
  147:
  148:     function [3:0] s1;
  149:         input [5:0] in;
  150:         reg [3:0] s;
  151:         begin
  152:             case ({in[0],in[5]})
  153:                 0: begin
  154:                     case ({in[1],in[2],in[3],in[4]})
  155:                         0: s = 14;  1: s = 4;   2: s = 13;  3: s = 1;
  156:                         4: s = 2;   5: s = 15;  6: s = 11;  7: s = 8;
  157:                         8: s = 3;   9: s = 10;  10: s = 6;  11: s = 12;
  158:                         12: s = 5;  13: s = 9;  14: s = 0;  15: s = 7;
  159:                     endcase
  160:                 end
  161:                 1: begin
  162:                     case ({in[1],in[2],in[3],in[4]})
```

```
163:                        0: s = 0;   1: s = 15;  2: s = 7;   3: s = 4;
164:                        4: s = 14;  5: s = 2;   6: s = 13;  7: s = 1;
165:                        8: s = 10;  9: s = 6;   10: s = 12;     11: s = 11;
166:                        12: s = 9;  13: s = 5;  14: s = 3;  15: s = 8;
167:                    endcase
168:                end
169:                2: begin
170:                    case ({in[1],in[2],in[3],in[4]})
171:                        0: s = 4;   1: s = 1;   2: s = 14;  3: s = 8;
172:                        4: s = 13;  5: s = 6;   6: s = 2;   7: s = 11;
173:                        8: s = 15;  9: s = 12;  10: s = 9;  11: s = 7;
174:                        12: s = 3;  13: s = 10;     14: s = 5;  15: s = 0;
175:                    endcase
176:                end
177:                3: begin
178:                    case ({in[1],in[2],in[3],in[4]})
179:                        0: s = 15;  1: s = 12;  2: s = 8;   3: s = 2;
180:                        4: s = 4;   5: s = 9;   6: s = 1;   7: s = 7;
181:                        8: s = 5;   9: s = 11;  10: s = 3;  11: s = 14;
182:                        12: s = 10;     13: s = 0;  14: s = 6;  15: s = 13;
183:                    endcase
184:                end
185:            endcase
186:
187:            s1 = {s[0],s[1],s[2],s[3]};
188:        end
189:    endfunction
190:
191:    function [3:0] s2;
192:        input [5:0] in;
193:        reg [3:0] s;
194:        begin
195:            case ({in[0],in[5]})
196:                0: begin
197:                    case ({in[1],in[2],in[3],in[4]})
198:                        0: s = 15;  1: s = 1;   2: s = 8;   3: s = 14;
199:                        4: s = 6;   5: s = 11;  6: s = 3;   7: s = 4;
200:                        8: s = 9;   9: s = 7;   10: s = 2;  11: s = 13;
201:                        12: s = 12;     13: s = 0;  14: s = 5;  15: s = 10;
202:                    endcase
203:                end
204:                1: begin
205:                    case ({in[1],in[2],in[3],in[4]})
206:                        0: s = 3;   1: s = 13;  2: s = 4;   3: s = 7;
207:                        4: s = 15;  5: s = 2;   6: s = 8;   7: s = 14;
208:                        8: s = 12;  9: s = 0;   10: s = 1;  11: s = 10;
209:                        12: s = 6;  13: s = 9;  14: s = 11;     15: s = 5;
210:                    endcase
211:                end
212:                2: begin
213:                    case ({in[1],in[2],in[3],in[4]})
214:                        0: s = 0;   1: s = 14;  2: s = 7;   3: s = 11;
215:                        4: s = 10;  5: s = 4;   6: s = 13;  7: s = 1;
216:                        8: s = 5;   9: s = 8;   10: s = 12;     11: s = 6;
217:                        12: s = 9;  13: s = 3;  14: s = 2;  15: s = 15;
218:                    endcase
219:                end
220:                3: begin
221:                    case ({in[1],in[2],in[3],in[4]})
222:                        0: s = 13;  1: s = 8;   2: s = 10;  3: s = 1;
223:                        4: s = 3;   5: s = 15;  6: s = 4;   7: s = 2;
224:                        8: s = 11;  9: s = 6;   10: s = 7;  11: s = 12;
225:                        12: s = 0;  13: s = 5;  14: s = 14;     15: s = 9;
226:                    endcase
227:                end
228:            endcase
```

```
229:
230:             s2 = {s[0],s[1],s[2],s[3]};
231:         end
232:     endfunction
233:
234:     function [3:0] s3;
235:         input [5:0] in;
236:         reg [3:0] s;
237:         begin
238:             case ({in[0],in[5]})
239:                 0: begin
240:                     case ({in[1],in[2],in[3],in[4]})
241:                         0: s = 10;  1: s = 0;   2: s = 9;   3: s = 14;
242:                         4: s = 6;   5: s = 3;   6: s = 15;  7: s = 5;
243:                         8: s = 1;   9: s = 13;  10: s = 12;     11: s = 7;
244:                         12: s = 11;     13: s = 4;  14: s = 2;  15: s = 8;
245:                     endcase
246:                 end
247:                 1: begin
248:                     case ({in[1],in[2],in[3],in[4]})
249:                         0: s = 13;  1: s = 7;   2: s = 0;   3: s = 9;
250:                         4: s = 3;   5: s = 4;   6: s = 6;   7: s = 10;
251:                         8: s = 2;   9: s = 8;   10: s = 5;  11: s = 14;
252:                         12: s = 12;     13: s = 11;     14: s = 15;     15: s = 1;
253:                     endcase
254:                 end
255:                 2: begin
256:                     case ({in[1],in[2],in[3],in[4]})
257:                         0: s = 13;  1: s = 6;   2: s = 4;   3: s = 9;
258:                         4: s = 8;   5: s = 15;  6: s = 3;   7: s = 0;
259:                         8: s = 11;  9: s = 1;   10: s = 2;  11: s = 12;
260:                         12: s = 5;  13: s = 10;     14: s = 14;     15: s = 7;
261:                     endcase
262:                 end
263:                 3: begin
264:                     case ({in[1],in[2],in[3],in[4]})
265:                         0: s = 1;   1: s = 10;  2: s = 13;  3: s = 0;
266:                         4: s = 6;   5: s = 9;   6: s = 8;   7: s = 7;
267:                         8: s = 4;   9: s = 15;  10: s = 14;     11: s = 3;
268:                         12: s = 11;     13: s = 5;  14: s = 2;  15: s = 12;
269:                     endcase
270:                 end
271:             endcase
272:
273:             s3 = {s[0],s[1],s[2],s[3]};
274:         end
275:     endfunction
276:
277:     function [3:0] s4;
278:         input [5:0] in;
279:         reg [3:0] s;
280:         begin
281:             case ({in[0],in[5]})
282:                 0: begin
283:                     case ({in[1],in[2],in[3],in[4]})
284:                         0: s = 7;   1: s = 13;  2: s = 14;  3: s = 3;
285:                         4: s = 0;   5: s = 6;   6: s = 9;   7: s = 10;
286:                         8: s = 1;   9: s = 2;   10: s = 8;  11: s = 5;
287:                         12: s = 11;     13: s = 12;     14: s = 4;  15: s = 15;
288:                     endcase
289:                 end
290:                 1: begin
291:                     case ({in[1],in[2],in[3],in[4]})
292:                         0: s = 13;  1: s = 8;   2: s = 11;  3: s = 5;
293:                         4: s = 6;   5: s = 15;  6: s = 0;   7: s = 3;
294:                         8: s = 4;   9: s = 7;   10: s = 2;  11: s = 12;
```

```
295:                        12: s = 1;  13: s = 10;     14: s = 14;     15: s = 9;
296:                    endcase
297:                end
298:                2: begin
299:                    case ({in[1],in[2],in[3],in[4]})
300:                        0: s = 10;  1: s = 6;   2: s = 9;   3: s = 0;
301:                        4: s = 12;  5: s = 11;  6: s = 7;   7: s = 13;
302:                        8: s = 15;  9: s = 1;   10: s = 3;  11: s = 14;
303:                        12: s = 5;  13: s = 2;  14: s = 8;  15: s = 4;
304:                    endcase
305:                end
306:                3: begin
307:                    case ({in[1],in[2],in[3],in[4]})
308:                        0: s = 3;   1: s = 15;  2: s = 0;   3: s = 6;
309:                        4: s = 10;  5: s = 1;   6: s = 13;  7: s = 8;
310:                        8: s = 9;   9: s = 4;   10: s = 5;  11: s = 11;
311:                        12: s = 12;     13: s = 7;  14: s = 2;  15: s = 14;
312:                    endcase
313:                end
314:            endcase
315:
316:            s4 = {s[0],s[1],s[2],s[3]};
317:        end
318:    endfunction
319:
320:    function [3:0] s5;
321:        input [5:0] in;
322:        reg [3:0] s;
323:        begin
324:            case ({in[0],in[5]})
325:                0: begin
326:                    case ({in[1],in[2],in[3],in[4]})
327:                        0: s = 2;   1: s = 12;  2: s = 4;   3: s = 1;
328:                        4: s = 7;   5: s = 10;  6: s = 11;  7: s = 6;
329:                        8: s = 8;   9: s = 5;   10: s = 3;  11: s = 15;
330:                        12: s = 13;     13: s = 0;  14: s = 14;     15: s = 9;
331:                    endcase
332:                end
333:                1: begin
334:                    case ({in[1],in[2],in[3],in[4]})
335:                        0: s = 14;  1: s = 11;  2: s = 2;   3: s = 12;
336:                        4: s = 4;   5: s = 7;   6: s = 13;  7: s = 1;
337:                        8: s = 5;   9: s = 0;   10: s = 15;     11: s = 10;
338:                        12: s = 3;  13: s = 9;  14: s = 8;  15: s = 6;
339:                    endcase
340:                end
341:                2: begin
342:                    case ({in[1],in[2],in[3],in[4]})
343:                        0: s = 4;   1: s = 2;   2: s = 1;   3: s = 11;
344:                        4: s = 10;  5: s = 13;  6: s = 7;   7: s = 8;
345:                        8: s = 15;  9: s = 9;   10: s = 12;     11: s = 5;
346:                        12: s = 6;  13: s = 3;  14: s = 0;  15: s = 14;
347:                    endcase
348:                end
349:                3: begin
350:                    case ({in[1],in[2],in[3],in[4]})
351:                        0: s = 11;  1: s = 8;   2: s = 12;  3: s = 7;
352:                        4: s = 1;   5: s = 14;  6: s = 2;   7: s = 13;
353:                        8: s = 6;   9: s = 15;  10: s = 0;  11: s = 9;
354:                        12: s = 10;     13: s = 4;  14: s = 5;  15: s = 3;
355:                    endcase
356:                end
357:            endcase
358:
359:            s5 = {s[0],s[1],s[2],s[3]};
360:        end
```

```
361:     endfunction
362:
363:     function [3:0] s6;
364:         input [5:0] in;
365:         reg [3:0] s;
366:         begin
367:             case ({in[0],in[5]})
368:                 0: begin
369:                     case ({in[1],in[2],in[3],in[4]})
370:                         0: s = 12;  1: s = 1;   2: s = 10;  3: s = 15;
371:                         4: s = 9;   5: s = 2;   6: s = 6;   7: s = 8;
372:                         8: s = 0;   9: s = 13;  10: s = 3;  11: s = 4;
373:                         12: s = 14;     13: s = 7;  14: s = 5;  15: s = 11;
374:                     endcase
375:                 end
376:                 1: begin
377:                     case ({in[1],in[2],in[3],in[4]})
378:                         0: s = 10;  1: s = 15;  2: s = 4;   3: s = 2;
379:                         4: s = 7;   5: s = 12;  6: s = 9;   7: s = 5;
380:                         8: s = 6;   9: s = 1;   10: s = 13;     11: s = 14;
381:                         12: s = 0;  13: s = 11;     14: s = 3;  15: s = 8;
382:                     endcase
383:                 end
384:                 2: begin
385:                     case ({in[1],in[2],in[3],in[4]})
386:                         0: s = 9;   1: s = 14;  2: s = 15;  3: s = 5;
387:                         4: s = 2;   5: s = 8;   6: s = 12;  7: s = 3;
388:                         8: s = 7;   9: s = 0;   10: s = 4;  11: s = 10;
389:                         12: s = 1;  13: s = 13;     14: s = 11;     15: s = 6;
390:                     endcase
391:                 end
392:                 3: begin
393:                     case ({in[1],in[2],in[3],in[4]})
394:                         0: s = 4;   1: s = 3;   2: s = 2;   3: s = 12;
395:                         4: s = 9;   5: s = 5;   6: s = 15;  7: s = 10;
396:                         8: s = 11;  9: s = 14;  10: s = 1;  11: s = 7;
397:                         12: s = 6;  13: s = 0;  14: s = 8;  15: s = 13;
398:                     endcase
399:                 end
400:             endcase
401:
402:             s6 = {s[0],s[1],s[2],s[3]};
403:         end
404:     endfunction
405:
406:     function [3:0] s7;
407:         input [5:0] in;
408:         reg [3:0] s;
409:         begin
410:             case ({in[0],in[5]})
411:                 0: begin
412:                     case ({in[1],in[2],in[3],in[4]})
413:                         0: s = 4;   1: s = 11;  2: s = 2;   3: s = 14;
414:                         4: s = 15;  5: s = 0;   6: s = 8;   7: s = 13;
415:                         8: s = 3;   9: s = 12;  10: s = 9;  11: s = 7;
416:                         12: s = 5;  13: s = 10;     14: s = 6;  15: s = 1;
417:                     endcase
418:                 end
419:                 1: begin
420:                     case ({in[1],in[2],in[3],in[4]})
421:                         0: s = 13;  1: s = 0;   2: s = 11;  3: s = 7;
422:                         4: s = 4;   5: s = 9;   6: s = 1;   7: s = 10;
423:                         8: s = 14;  9: s = 3;   10: s = 5;  11: s = 12;
424:                         12: s = 2;  13: s = 15;     14: s = 8;  15: s = 6;
425:                     endcase
426:                 end
```

```
427:                 2: begin
428:                     case ({in[1],in[2],in[3],in[4]})
429:                         0: s = 1;   1: s = 4;   2: s = 11;  3: s = 13;
430:                         4: s = 12;  5: s = 3;   6: s = 7;   7: s = 14;
431:                         8: s = 10;  9: s = 15;  10: s = 6;  11: s = 8;
432:                         12: s = 0;  13: s = 5;  14: s = 9;  15: s = 2;
433:                     endcase
434:                 end
435:                 3: begin
436:                     case ({in[1],in[2],in[3],in[4]})
437:                         0: s = 6;   1: s = 11;  2: s = 13;  3: s = 8;
438:                         4: s = 1;   5: s = 4;   6: s = 10;  7: s = 7;
439:                         8: s = 9;   9: s = 5;   10: s = 0;  11: s = 15;
440:                         12: s = 14;     13: s = 2;  14: s = 3;  15: s = 12;
441:                     endcase
442:                 end
443:             endcase
444:
445:             s7 = {s[0],s[1],s[2],s[3]};
446:         end
447:     endfunction
448:
449:     function [3:0] s8;
450:         input [5:0] in;
451:         reg [3:0] s;
452:         begin
453:             case ({in[0],in[5]})
454:                 0: begin
455:                     case ({in[1],in[2],in[3],in[4]})
456:                         0: s = 13;  1: s = 2;   2: s = 8;   3: s = 4;
457:                         4: s = 6;   5: s = 15;  6: s = 11;  7: s = 1;
458:                         8: s = 10;  9: s = 9;   10: s = 3;  11: s = 14;
459:                         12: s = 5;  13: s = 0;  14: s = 12;     15: s = 7;
460:                     endcase
461:                 end
462:                 1: begin
463:                     case ({in[1],in[2],in[3],in[4]})
464:                         0: s = 1;   1: s = 15;  2: s = 13;  3: s = 8;
465:                         4: s = 10;  5: s = 3;   6: s = 7;   7: s = 4;
466:                         8: s = 12;  9: s = 5;   10: s = 6;  11: s = 11;
467:                         12: s = 0;  13: s = 14;     14: s = 9;  15: s = 2;
468:                     endcase
469:                 end
470:                 2: begin
471:                     case ({in[1],in[2],in[3],in[4]})
472:                         0: s = 7;   1: s = 11;  2: s = 4;   3: s = 1;
473:                         4: s = 9;   5: s = 12;  6: s = 14;  7: s = 2;
474:                         8: s = 0;   9: s = 6;   10: s = 10;     11: s = 13;
475:                         12: s = 15;     13: s = 3;  14: s = 5;  15: s = 8;
476:                     endcase
477:                 end
478:                 3: begin
479:                     case ({in[1],in[2],in[3],in[4]})
480:                         0: s = 2;   1: s = 1;   2: s = 14;  3: s = 7;
481:                         4: s = 4;   5: s = 10;  6: s = 8;   7: s = 13;
482:                         8: s = 15;  9: s = 12;  10: s = 9;  11: s = 0;
483:                         12: s = 3;  13: s = 5;  14: s = 6;  15: s = 11;
484:                     endcase
485:                 end
486:             endcase
487:
488:             s8 = {s[0],s[1],s[2],s[3]};
489:         end
490:     endfunction
491:
492:
```

```
  493:     function [47:0] e2;
  494:         input [32:1] in;
  495:
  496:         e2 = {in[1],in[32:28],in[29:24],in[25:20],in[21:16],in[17:12],in[13:8],in[9:
4],in[5:1],in[32]};
  497:     endfunction
  498:
  499:     function [47:0] PC2;
  500:         input [56:1] in;
  501:
  502:         PC2 = {in[32],in[29],in[36],in[50],in[42],in[46],
  503:                in[53],in[34],in[56],in[39],in[49],in[44],
  504:                in[48],in[33],in[45],in[51],in[40],in[30],
  505:                in[55],in[47],in[37],in[31],in[52],in[41],
  506:
  507:                in[2],in[13],in[20],in[27],in[7],in[16],
  508:                in[8],in[26],in[4],in[12],in[19],in[23],
  509:                in[10],in[21],in[6],in[15],in[28],in[3],
  510:                in[5],in[1],in[24],in[11],in[17],in[14] };
  511:     endfunction
  512:
  513: endmodule
```