```
1   library ieee;
2
3   use ieee.std_logic_1164.all;
4   use ieee.std_logic_arith.all;
5   use ieee.std_logic_unsigned.all;
6
7   entity ControlUnit is
8      port (
9         clk, mclk: in std_logic;
10        enable: in std_logic;
11        statusC, statusZ: in std_logic;
12        inst: in std_logic_vector(31 downto 0);
13        A_mux, B_mux, IM_Mux1, REG_Mux : out std_logic;
14        IM_mux2, DATA_Mux: out std_logic_vector (1 downto 0);
15        ALU_op: out std_logic_vector(2 downto 0);
16        inc_PC, ld_PC, clr_IR, ld_IR: out std_logic;
17        clr_A, clr_B, clr_C, clr_Z : out std_logic;
18        ld_A, ld_b, ld_C, ld_Z: out std_logic;
19        T : out std_logic_vector(2 downto 0);
20        wen, en: out std_logic);
21  end ControlUnit;
22
23  architecture description of ControlUnit is
24  --define state types
25     type statetype is (state_0, state_1, state_2, s0);
26     --state signals
27     signal present_state: statetype;
28     signal inst_sig, inst_sig2: std_logic_vector(3 downto 0);
29        begin
30           inst_sig <= inst(31 downto 28);
31           inst_sig2 <= inst(27 downto 24);
32
33           --state machine
34           process(clk, enable, mclk, inst, statusC, statusZ)
35              begin
36                 if (enable = '0') then
37                    present_state <= s0;
38                 elsif (rising_edge(clk)) then
39                    case present_state is
40                       when s0 => present_state <= state_0;
41                       when state_0 => present_state <= state_1;
42                       when state_1 => present_state <= state_2;
43                       when state_2 => present_state <= state_0;
44                    end case;
45                 end if;
46           end process;
47
48           --decoder and memory signal
49              process(present_state)
50                 begin
51                    case present_state is
52                       when s0 => clr_IR <= '0';
53                                  ld_IR <= '1';
54                                  ld_PC <= '1';
55                                  inc_PC <= '0';
56                                  clr_A <= '0';
57                                  ld_A <= '0';
58                                  clr_B <= '0';
59                                  ld_b <= '0';
60                                  ld_C <= '0';
61                                  clr_C <= '0';
62                                  clr_Z <= '0';
```

```
63                                      ld_Z <= '0';
64                                      alu_op <= "XXX";
65                                      en <= 'X';
66                                      wen <= 'X';
67                                      A_mux <= 'X';
68                                      B_mux <= 'X';
69                                      REG_Mux <= 'X';
70                                      DATA_Mux <= "00";
71                                      IM_Mux1 <= 'X';
72                                      IM_mux2 <= "XX";
73
74                              when state_0 => clr_IR <= '0';
75                                      ld_IR <= '1';
76                                      ld_PC <= '1';
77                                      inc_PC <= '0';
78                                      clr_A <= '0';
79                                      ld_A <= '0';
80                                      clr_B <= '0';
81                                      ld_b <= '0';
82                                      ld_C <= '0';
83                                      clr_C <= '0';
84                                      clr_Z <= '0';
85                                      ld_Z <= '0';
86                                      alu_op <= "XXX";
87                                      en <= 'X';
88                                      wen <= 'X';
89                                      A_mux <= 'X';
90                                      B_mux <= 'X';
91                                      REG_Mux <= 'X';
92                                      DATA_Mux <= "00";
93                                      IM_Mux1 <= 'X';
94                                      IM_mux2 <= "XX";
95
96                              when state_1 => clr_IR <= '0';
97                                      ld_IR <= '0';
98                                      ld_PC <= '1';
99                                      inc_PC <= '1';
100                                     clr_A <= '0';
101                                     ld_A <= '0';
102                                     clr_B <= '0';
103                                     ld_b <= '0';
104                                     ld_C <= '0';
105                                     clr_C <= '0';
106                                     clr_Z <= '0';
107                                     ld_Z <= '0';
108                                     alu_op <= "XXX";
109                                     en <= 'X';
110                                     wen <= 'X';
111                                     A_mux <= 'X';
112                                     B_mux <= 'X';
113                                     REG_Mux <= 'X';
114                                     DATA_Mux <= "00";
115                                     IM_Mux1 <= 'X';
116                                     IM_mux2 <= "XX";
117
118                             case inst_sig is --sta
119                                   when "0010" =>
120                                     clr_IR <= '0';
121                                     ld_IR <= '0';
122                                     ld_PC <= '1';
123                                     inc_PC <= '1';
124                                     clr_A <= '0';
```

```
125                     ld_A <= '0';
126                     clr_B <= '0';
127                     ld_b <= '0';
128                     ld_C <= '0';
129                     clr_C <= '0';
130                     clr_Z <= '0';
131                     ld_Z <= '0';
132                     alu_op <= "XXX";
133                     en <= '1';
134                     wen <= '1';
135                     A_mux <= 'X';
136                     B_mux <= 'X';
137                     REG_Mux <= 'X';
138                     DATA_Mux <= "00";
139                     IM_Mux1 <= 'X';
140                     IM_mux2 <= "XX";
141 
142                   when "0011" => --stb
143                     clr_IR <= '0';
144                     ld_IR <= '0';
145                     ld_PC <= '1';
146                     inc_PC <= '1';
147                     clr_A <= '0';
148                     ld_A <= '0';
149                     clr_B <= '0';
150                     ld_b <= '0';
151                     ld_C <= '0';
152                     clr_C <= '0';
153                     clr_Z <= '0';
154                     ld_Z <= '0';
155                     alu_op <= "XXX";
156                     en <= '1';
157                     wen <= '1';
158                     A_mux <= 'X';
159                     B_mux <= 'X';
160                     REG_Mux <= '1';
161                     DATA_Mux <= "00";
162                     IM_Mux1 <= 'X';
163                     IM_mux2 <= "XX";
164 
165                   when "1001" => --lda
166                     clr_IR <= '0';
167                     ld_IR <= '0';
168                     ld_PC <= '1';
169                     inc_PC <= '1';
170                     clr_A <= '0';
171                     ld_A <= '1';
172                     clr_B <= '0';
173                     ld_b <= '0';
174                     ld_C <= '0';
175                     clr_C <= '0';
176                     clr_Z <= '0';
177                     ld_Z <= '0';
178                     alu_op <= "XXX";
179                     en <= '1';
180                     wen <= '0';
181                     A_mux <= 'X';
182                     B_mux <= 'X';
183                     REG_Mux <= '1';
184                     DATA_Mux <= "01";
185                     IM_Mux1 <= 'X';
186                     IM_mux2 <= "XX";
```

```
187
188                         when "1010" => --ldb
189                           clr_IR <= '0';
190                           ld_IR <= '0';
191                           ld_PC <= '1';
192                           inc_PC <= '1';
193                           clr_A <= '0';
194                           ld_A <= '0';
195                           clr_B <= '0';
196                           ld_b <= '1';
197                           ld_C <= '0';
198                           clr_C <= '0';
199                           clr_Z <= '0';
200                           ld_Z <= '0';
201                           alu_op <= "XXX";
202                           en <= '1';
203                           wen <= '0';
204                           A_mux <= 'X';
205                           B_mux <= 'X';
206                           REG_Mux <= '1';
207                           DATA_Mux <= "01";
208                           IM_Mux1 <= 'X';
209                           IM_mux2 <= "XX";
210
211                         when others =>
212                            clr_IR <= '0';
213                      end case; --inst_sig end
214
215                when state_2 =>
216                   case inst_sig is
217                      when "0000" => --ldbi
218                        clr_IR <= '0';
219                        ld_IR <= '0';
220                        ld_PC <= '0';
221                        inc_PC <= '0';
222                        clr_A <= '0';
223                        ld_A <= '1';
224                        clr_B <= '0';
225                        ld_b <= '0';
226                        ld_C <= '0';
227                        clr_C <= '0';
228                        clr_Z <= '0';
229                        ld_Z <= '0';
230                        alu_op <= "XXX";
231                        en <= 'X';
232                        wen <= 'X';
233                        A_mux <= '1';
234                        B_mux <= 'X';
235                        REG_Mux <= 'X';
236                        DATA_Mux <= "XX";
237                        IM_Mux1 <= 'X';
238                        IM_mux2 <= "XX";
239
240                      when "0001" => --sta
241                        clr_IR <= '0';
242                        ld_IR <= '0';
243                        ld_PC <= '0';
244                        inc_PC <= '0';
245                        clr_A <= '0';
246                        ld_A <= '0';
247                        clr_B <= '0';
248                        ld_b <= '1';
```

```
249                     ld_C <= '0';
250                     clr_C <= '0';
251                     clr_Z <= '0';
252                     ld_Z <= '0';
253                     alu_op <= "XXX";
254                     en <= 'X';
255                     wen <= 'X';
256                     A_mux <= 'X';
257                     B_mux <= '1';
258                     REG_Mux <= 'X';
259                     DATA_Mux <= "XX";
260                     IM_Mux1 <= 'X';
261                     IM_mux2 <= "XX";
262 
263                   when "0010" => --stb
264                     clr_IR <= '0';
265                     ld_IR <= '0';
266                     ld_PC <= '0';
267                     inc_PC <= '0';
268                     clr_A <= '0';
269                     ld_A <= '0';
270                     clr_B <= '0';
271                     ld_b <= '1';
272                     ld_C <= '0';
273                     clr_C <= '0';
274                     clr_Z <= '0';
275                     ld_Z <= '0';
276                     alu_op <= "XXX";
277                     en <= '1';
278                     wen <= '1';
279                     A_mux <= 'X';
280                     B_mux <= 'X';
281                     REG_Mux <= 'X';
282                     DATA_Mux <= "XX";
283                     IM_Mux1 <= 'X';
284                     IM_mux2 <= "XX";
285 
286                   when "0011" => --lda
287                     clr_IR <= '0';
288                     ld_IR <= '0';
289                     ld_PC <= '0';
290                     inc_PC <= '0';
291                     clr_A <= '0';
292                     ld_A <= '0';
293                     clr_B <= '0';
294                     ld_b <= '1';
295                     ld_C <= '0';
296                     clr_C <= '0';
297                     clr_Z <= '0';
298                     ld_Z <= '0';
299                     alu_op <= "XXX";
300                     en <= '1';
301                     wen <= '1';
302                     A_mux <= 'X';
303                     B_mux <= 'X';
304                     REG_Mux <= '1';
305                     DATA_Mux <= "XX";
306                     IM_Mux1 <= 'X';
307                     IM_mux2 <= "XX";
308 
309                   when "1001" => --ldab
310                     clr_IR <= '0';
```

```
311                     ld_IR <= '0';
312                     ld_PC <= '0';
313                     inc_PC <= '0';
314                     clr_A <= '0';
315                     ld_A <= '1';
316                     clr_B <= '0';
317                     ld_b <= '0';
318                     ld_C <= '0';
319                     clr_C <= '0';
320                     clr_Z <= '0';
321                     ld_Z <= '0';
322                     alu_op <= "XXX";
323                     en <= '1';
324                     wen <= '0';
325                     A_mux <= '0';
326                     B_mux <= 'X';
327                     REG_Mux <= '1';
328                     DATA_Mux <= "01";
329                     IM_Mux1 <= 'X';
330                     IM_mux2 <= "XX";
331 
332                   when "1010" => --lui
333                     clr_IR <= '0';
334                     ld_IR <= '0';
335                     ld_PC <= '0';
336                     inc_PC <= '0';
337                     clr_A <= '0';
338                     ld_A <= '0';
339                     clr_B <= '0';
340                     ld_b <= '1';
341                     ld_C <= '0';
342                     clr_C <= '0';
343                     clr_Z <= '0';
344                     ld_Z <= '0';
345                     alu_op <= "XXX";
346                     en <= '1';
347                     wen <= '0';
348                     A_mux <= '0';
349                     B_mux <= 'X';
350                     REG_Mux <= 'X';
351                     DATA_Mux <= "01";
352                     IM_Mux1 <= 'X';
353                     IM_mux2 <= "XX";
354 
355                   when "0100" => --jmp
356                     clr_IR <= '0';
357                     ld_IR <= '0';
358                     ld_PC <= '0';
359                     inc_PC <= '0';
360                     clr_A <= '1';
361                     ld_A <= '0';
362                     clr_B <= '0';
363                     ld_b <= '0';
364                     ld_C <= '0';
365                     clr_C <= '0';
366                     clr_Z <= '0';
367                     ld_Z <= '0';
368                     alu_op <= "001";
369                     en <= 'X';
370                     wen <= 'X';
371                     A_mux <= '0';
372                     B_mux <= 'X';
```

```
373                     REG_Mux <= 'X';
374                     DATA_Mux <= "10";
375                     IM_Mux1 <= '1';
376                     IM_mux2 <= "XX";
377
378                   when "0101" => --bne
379                     clr_IR <= '0';
380                     ld_IR <= '0';
381                     ld_PC <= '1';
382                     inc_PC <= '0';
383                     clr_A <= '0';
384                     ld_A <= '0';
385                     clr_B <= '0';
386                     ld_b <= '0';
387                     ld_C <= '0';
388                     clr_C <= '0';
389                     clr_Z <= '0';
390                     ld_Z <= '0';
391                     alu_op <= "XXX";
392                     en <= 'X';
393                     wen <= 'X';
394                     A_mux <= '0';
395                     B_mux <= 'X';
396                     REG_Mux <= 'X';
397                     DATA_Mux <= "XX";
398                     IM_Mux1 <= 'X';
399                     IM_mux2 <= "XX";
400
401                    when "1000" => --beq
402                     clr_IR <= '0';
403                     ld_IR <= '0';
404                     ld_PC <= '0';
405                     inc_PC <= '0';
406                     clr_A <= '0';
407                     ld_A <= '0';
408                     clr_B <= '0';
409                     ld_b <= '0';
410                     ld_C <= '0';
411                     clr_C <= '0';
412                     clr_Z <= '0';
413                     ld_Z <= '1';
414                     alu_op <= "110";
415                     en <= 'X';
416                     wen <= 'X';
417                     A_mux <= '0';
418                     B_mux <= 'X';
419                     REG_Mux <= 'X';
420                     DATA_Mux <= "XX";
421                     IM_Mux1 <= '0';
422                     IM_mux2 <= "00";
423                      if statusZ = '0' then
424                         ld_PC <= '1';
425                      else
426                         ld_PC <= '0';
427                      end if;
428
429                   when "0110" => --add - check multiplexer A
430                     clr_IR <= '0';
431                     ld_IR <= '0';
432                     ld_PC <= '0';
433                     inc_PC <= '0';
434                     clr_A <= '0';
```

```
435                         ld_A <= '0';
436                         clr_B <= '0';
437                         ld_b <= '0';
438                         ld_C <= '0';
439                         clr_C <= '0';
440                         clr_Z <= '0';
441                         ld_Z <= '1';
442                         alu_op <= "110";
443                         en <= 'X';
444                         wen <= 'X';
445                         A_mux <= '0';
446                         B_mux <= 'X';
447                         REG_Mux <= 'X';
448                         DATA_Mux <= "XX";
449                         IM_Mux1 <= '0';
450                         IM_mux2 <= "00";
451                          if statusZ = '1' then
452                             ld_PC <= '1';
453                          else
454                             ld_PC <= '0';
455                          end if;
456
457                      when "0111" =>
458                         case inst_sig2 is
459                            when "0000" => --add
460                              clr_IR <= '0';
461                              ld_IR <= '0';
462                              ld_PC <= '0';
463                              inc_PC <= '0';
464                              clr_A <= '0';
465                              ld_A <= '1';
466                              clr_B <= '0';
467                              ld_b <= '0';
468                              ld_C <= '0';
469                              clr_C <= '0';
470                              clr_Z <= '1';
471                              ld_Z <= '0';
472                              alu_op <= "010";
473                              en <= 'X';
474                              wen <= 'X';
475                              A_mux <= '0';
476                              B_mux <= 'X';
477                              REG_Mux <= 'X';
478                              DATA_Mux <= "10";
479                              IM_Mux1 <= '0';
480                              IM_mux2 <= "00";
481
482                            when "0001" => --addi
483                              clr_IR <= '0';
484                              ld_IR <= '0';
485                              ld_PC <= '0';
486                              inc_PC <= '0';
487                              clr_A <= '0';
488                              ld_A <= '1';
489                              clr_B <= '0';
490                              ld_b <= '0';
491                              ld_C <= '1';
492                              clr_C <= '0';
493                              clr_Z <= '1';
494                              ld_Z <= '0';
495                              alu_op <= "010";
496                              en <= 'X';
```

```
497                 wen <= 'X';
498                 A_mux <= '0';
499                 B_mux <= 'X';
500                 REG_Mux <= 'X';
501                 DATA_Mux <= "10";
502                 IM_Mux1 <= '0';
503                 IM_mux2 <= "01";
504 
505               when "0010" => --sub
506                 clr_IR <= '0';
507                 ld_IR <= '0';
508                 ld_PC <= '0';
509                 inc_PC <= '0';
510                 clr_A <= '0';
511                 ld_A <= '1';
512                 clr_B <= '0';
513                 ld_b <= '0';
514                 ld_C <= '1';
515                 clr_C <= '0';
516                 clr_Z <= '1';
517                 ld_Z <= '0';
518                 alu_op <= "110";
519                 en <= 'X';
520                 wen <= 'X';
521                 A_mux <= '0';
522                 B_mux <= 'X';
523                 REG_Mux <= 'X';
524                 DATA_Mux <= "10";
525                 IM_Mux1 <= '0';
526                 IM_mux2 <= "00";
527 
528               when "0011" => --inca
529                 clr_IR <= '0';
530                 ld_IR <= '0';
531                 ld_PC <= '0';
532                 inc_PC <= '0';
533                 clr_A <= '0';
534                 ld_A <= '1';
535                 clr_B <= '0';
536                 ld_b <= '0';
537                 ld_C <= '1';
538                 clr_C <= '0';
539                 clr_Z <= '1';
540                 ld_Z <= '0';
541                 alu_op <= "010";
542                 en <= 'X';
543                 wen <= 'X';
544                 A_mux <= '0';
545                 B_mux <= 'X';
546                 REG_Mux <= 'X';
547                 DATA_Mux <= "10";
548                 IM_Mux1 <= '0';
549                 IM_mux2 <= "10";
550 
551               when "0100" => --rol
552                 clr_IR <= '0';
553                 ld_IR <= '0';
554                 ld_PC <= '0';
555                 inc_PC <= '0';
556                 clr_A <= '0';
557                 ld_A <= '1';
558                 clr_B <= '0';
```

```
559                                 ld_b <= '0';
560                                 ld_C <= '1';
561                                 clr_C <= '0';
562                                 clr_Z <= '1';
563                                 ld_Z <= '0';
564                                 alu_op <= "100";
565                                 en <= 'X';
566                                 wen <= 'X';
567                                 A_mux <= '0';
568                                 B_mux <= 'X';
569                                 REG_Mux <= 'X';
570                                 DATA_Mux <= "10";
571                                 IM_Mux1 <= '0';
572                                 IM_mux2 <= "XX";
573 
574                             when "0101" => --clra
575                                 clr_IR <= '0';
576                                 ld_IR <= '0';
577                                 ld_PC <= '0';
578                                 inc_PC <= '0';
579                                 clr_A <= '0';
580                                 ld_A <= '1';
581                                 clr_B <= '0';
582                                 ld_b <= '0';
583                                 ld_C <= '0';
584                                 clr_C <= '0';
585                                 clr_Z <= '0';
586                                 ld_Z <= '0';
587                                 alu_op <= "XXX";
588                                 en <= 'X';
589                                 wen <= 'X';
590                                 A_mux <= 'X';
591                                 B_mux <= 'X';
592                                 REG_Mux <= 'X';
593                                 DATA_Mux <= "XX";
594                                 IM_Mux1 <= 'X';
595                                 IM_mux2 <= "XX";
596 
597                             when "0110" => --clrb
598                                 clr_IR <= '0';
599                                 ld_IR <= '0';
600                                 ld_PC <= '0';
601                                 inc_PC <= '0';
602                                 clr_A <= '0';
603                                 ld_A <= '0';
604                                 clr_B <= '1';
605                                 ld_b <= '0';
606                                 ld_C <= '0';
607                                 clr_C <= '0';
608                                 clr_Z <= '0';
609                                 ld_Z <= '0';
610                                 alu_op <= "XXX";
611                                 en <= 'X';
612                                 wen <= 'X';
613                                 A_mux <= 'X';
614                                 B_mux <= 'X';
615                                 REG_Mux <= 'X';
616                                 DATA_Mux <= "XX";
617                                 IM_Mux1 <= 'X';
618                                 IM_mux2 <= "XX";
619 
620                             when "0111" => --clrc
```

```
621                 clr_IR <= '0';
622                 ld_IR <= '0';
623                 ld_PC <= '0';
624                 inc_PC <= '0';
625                 clr_A <= '0';
626                 ld_A <= '0';
627                 clr_B <= '0';
628                 ld_b <= '0';
629                 ld_C <= '0';
630                 clr_C <= '1';
631                 clr_Z <= '0';
632                 ld_Z <= '0';
633                 alu_op <= "XXX";
634                 en <= 'X';
635                 wen <= 'X';
636                 A_mux <= 'X';
637                 B_mux <= 'X';
638                 REG_Mux <= 'X';
639                 DATA_Mux <= "XX";
640                 IM_Mux1 <= 'X';
641                 IM_mux2 <= "XX";
642 
643               when "1000" => --clrz
644                 clr_IR <= '0';
645                 ld_IR <= '0';
646                 ld_PC <= '0';
647                 inc_PC <= '0';
648                 clr_A <= '0';
649                 ld_A <= '0';
650                 clr_B <= '0';
651                 ld_b <= '0';
652                 ld_C <= '0';
653                 clr_C <= '0';
654                 clr_Z <= '1';
655                 ld_Z <= '0';
656                 alu_op <= "XXX";
657                 en <= 'X';
658                 wen <= 'X';
659                 A_mux <= 'X';
660                 B_mux <= 'X';
661                 REG_Mux <= 'X';
662                 DATA_Mux <= "XX";
663                 IM_Mux1 <= 'X';
664                 IM_mux2 <= "XX";
665 
666               when "1001" => --andi
667                 clr_IR <= '0';
668                 ld_IR <= '0';
669                 ld_PC <= '0';
670                 inc_PC <= '0';
671                 clr_A <= '0';
672                 ld_A <= '1';
673                 clr_B <= '0';
674                 ld_b <= '0';
675                 ld_C <= '1';
676                 clr_C <= '0';
677                 clr_Z <= '0';
678                 ld_Z <= '1';
679                 alu_op <= "000";
680                 en <= 'X';
681                 wen <= 'X';
682                 A_mux <= 'X';
```

```
683                 B_mux <= 'X';
684                 REG_Mux <= 'X';
685                 DATA_Mux <= "10";
686                 IM_Mux1 <= '0';
687                 IM_mux2 <= "01";
688 
689               when "1011" => --and
690                 clr_IR <= '0';
691                 ld_IR <= '0';
692                 ld_PC <= '0';
693                 inc_PC <= '0';
694                 clr_A <= '0';
695                 ld_A <= '1';
696                 clr_B <= '0';
697                 ld_b <= '0';
698                 ld_C <= '1';
699                 clr_C <= '0';
700                 clr_Z <= '0';
701                 ld_Z <= '1';
702                 alu_op <= "000";
703                 en <= 'X';
704                 wen <= 'X';
705                 A_mux <= '0';
706                 B_mux <= 'X';
707                 REG_Mux <= 'X';
708                 DATA_Mux <= "10";
709                 IM_Mux1 <= '0';
710                 IM_mux2 <= "00";
711 
712               when "1101" => --ori
713                 clr_IR <= '0';
714                 ld_IR <= '0';
715                 ld_PC <= '0';
716                 inc_PC <= '0';
717                 clr_A <= '0';
718                 ld_A <= '1';
719                 clr_B <= '0';
720                 ld_b <= '0';
721                 ld_C <= '1';
722                 clr_C <= '0';
723                 clr_Z <= '0';
724                 ld_Z <= '1';
725                 alu_op <= "001";
726                 en <= 'X';
727                 wen <= 'X';
728                 A_mux <= '0';
729                 B_mux <= 'X';
730                 REG_Mux <= 'X';
731                 DATA_Mux <= "10";
732                 IM_Mux1 <= '0';
733                 IM_mux2 <= "01";
734 
735               when "1110" => --deca
736                 clr_IR <= '0';
737                 ld_IR <= '0';
738                 ld_PC <= '0';
739                 inc_PC <= '0';
740                 clr_A <= '0';
741                 ld_A <= '1';
742                 clr_B <= '0';
743                 ld_b <= '0';
744                 ld_C <= '1';
```

```
                                    clr_C <= '0';
                                    clr_Z <= '0';
                                    ld_Z <= '1';
                                    alu_op <= "110";
                                    en <= 'X';
                                    wen <= 'X';
                                    A_mux <= '0';
                                    B_mux <= 'X';
                                    REG_Mux <= 'X';
                                    DATA_Mux <= "10";
                                    IM_Mux1 <= '0';
                                    IM_mux2 <= "10";

                                  when "1111" => --ror
                                    clr_IR <= '0';
                                    ld_IR <= '0';
                                    ld_PC <= '0';
                                    inc_PC <= '0';
                                    clr_A <= '0';
                                    ld_A <= '1';
                                    clr_B <= '0';
                                    ld_b <= '0';
                                    ld_C <= '1';
                                    clr_C <= '0';
                                    clr_Z <= '0';
                                    ld_Z <= '1';
                                    alu_op <= "101";
                                    en <= 'X';
                                    wen <= 'X';
                                    A_mux <= '0';
                                    B_mux <= 'X';
                                    REG_Mux <= 'X';
                                    DATA_Mux <= "10";
                                    IM_Mux1 <= '0';
                                    IM_mux2 <= "XX";

                                  when others =>
                                     clr_IR <= '0';
                                  end case;
                               when others =>
                                  clr_IR <= '0';
                            end case;
                         end case;
                end process;
        with present_state select
           T <= "000" when s0,
                "001" when state_0,
                "010" when state_1,
                "011" when state_2,
                "111" when others;
   end description;
```