```
1   library ieee;
2   use ieee.std_logic_1164.all;
3
4   entity Control_New is
5   port(
6   clk, mclk : in std_logic;
7   enable : in std_logic;
8   statusC, statusZ : in std_logic;
9   INST : in std_logic_vector(31 downto 0);
10  A_Mux, B_Mux : out std_logic;
11  IM_MUX1, REG_MUX : out std_logic;
12  IM_MUX2, DATA_Mux : out std_logic_vector(1 downto 0);
13  ALU_op : out std_logic_vector(2 downto 0);
14  inc_PC, ld_PC : out std_logic;
15  clr_IR, ld_IR : out std_logic;
16  clr_A, clr_B, clr_C, clr_Z : out std_logic;
17  ld_A, ld_B, ld_C, ld_Z : out std_logic;
18  T : out std_logic_vector(2 downto 0);
19  wen, en : out std_logic
20  );
21  end Control_New;
22
23  architecture description of Control_New is
24     type STATETYPE is (state_0, state_1, state_2);
25     signal present_state : STATETYPE;
26     signal Instruction_sig : std_logic_vector(3 downto 0);
27     signal Instruction_sig2 : std_logic_vector(7 downto 0);
28  begin
29     Instruction_sig <= INST(31 downto 28);
30     Instruction_sig2 <= INST(31 downto 24);
31
32  -----OPERATION DECODER-----
33  process (present_state, INST, statusC, statusZ, enable, Instruction_sig,
    Instruction_sig2)
34  begin
35     if enable = '1' then
36        if present_state = state_0 then
37           DATA_Mux <= "00"; --Fetch address of the next instruction_sig
38           clr_IR<='0';
39           ld_IR<='1';
40           ld_PC<='0';
41           inc_PC<='0';
42           clr_A<='0';
43           ld_A<='0';
44           ld_B<='0';
45           clr_B<='0';
46           clr_C<='0';
47           ld_C<='0';
48           clr_Z<='0';
49           ld_Z<='0';
50           en<='0';
51           wen<='0';
52
53        elsif present_state = state_1 then
54           clr_IR<='0';
55           ld_IR<='0';
56           ld_PC<='1';
57           inc_PC<='1';
58           clr_A<='0';
59           ld_A<='0';
60           ld_B<='0';
61           clr_B<='0';
62           clr_C<='0';
63           ld_C<='0';
64           clr_Z<='0';
65           ld_Z<='0';
```

```
66              en<='0';
67              wen<='0';
68
69              if Instruction_sig = "0010" then --STA
70              clr_IR<='0';
71              ld_IR<='0';
72              ld_PC<='1';
73              inc_PC<='1';
74              clr_A<='0';
75              ld_A<='0';
76              ld_B<='0';
77              clr_B<='0';
78              clr_C<='0';
79              ld_C<='0';
80              clr_Z<='0';
81              ld_Z<='0';
82              REG_MUX<='0';
83              DATA_Mux<="00";
84              en<='1';
85              wen<='1';
86
87              elsif Instruction_sig = "0011" then --STB
88              clr_IR<='0';
89              ld_IR<='0';
90              ld_PC<='1';
91              inc_PC<='1';
92              clr_A<='0';
93              ld_A<='0';
94              ld_B<='0';
95              clr_B<='0';
96              clr_C<='0';
97              ld_C<='0';
98              clr_Z<='0';
99              ld_Z<='0';
100             REG_MUX<='1';
101             DATA_Mux<="00";
102             en<='1';
103             wen<='1';
104
105             elsif Instruction_sig = "1001" then --LDA
106             clr_IR<='0';
107             ld_IR<='0';
108             ld_PC<='1';
109             inc_PC<='1';
110             clr_A<='0';
111             ld_A<='1';
112             ld_B<='0';
113             clr_B<='0';
114             clr_C<='0';
115             ld_C<='0';
116             clr_Z<='0';
117             ld_Z<='0';
118             A_Mux<='0';
119             DATA_Mux<="01";
120             en<='1';
121             wen<='0';
122
123             elsif Instruction_sig = "1010" then --LDB
124             clr_IR<='0';
125             ld_IR<='0';
126             ld_PC<='1';
127             inc_PC<='1';
128             clr_A<='0';
129             ld_A<='0';
130             ld_B<='1';
131             clr_B<='0';
```

```
132             clr_C<='0';
133             ld_C<='0';
134             clr_Z<='0';
135             ld_Z<='0';
136             B_Mux<='0';
137             DATA_Mux<="01";
138             en<='1';
139             wen<='1';
140         end if; --end if for load store in stage 1
141 
142         elsif present_state = state_2 then
143 
144             if Instruction_sig = "0101" then --JUMP
145             clr_IR<='0';
146             ld_IR<='0';
147             ld_PC<='1';
148             inc_PC<='0';
149             clr_A<='0';
150             ld_A<='0';
151             ld_B<='0';
152             clr_B<='0';
153             clr_C<='0';
154             ld_C<='0';
155             clr_Z<='0';
156             ld_Z<='0';
157 
158             elsif Instruction_sig = "0110" then --BEQ
159             clr_IR<='0';
160             ld_IR<='0';
161             ld_PC<='1';
162             inc_PC<='0';
163             clr_A<='0';
164             ld_A<='0';
165             ld_B<='0';
166             clr_B<='0';
167             clr_C<='0';
168             ld_C<='0';
169             clr_Z<='0';
170             ld_Z<='0';
171 
172             elsif Instruction_sig = "1000" then --BNE
173             clr_IR<='0';
174             ld_IR<='0';
175             ld_PC<='1';
176             inc_PC<='0';
177             clr_A<='0';
178             ld_A<='0';
179             ld_B<='0';
180             clr_B<='0';
181             clr_C<='0';
182             ld_C<='0';
183             clr_Z<='0';
184             ld_Z<='0';
185 
186             elsif Instruction_sig = "1001" then --LDA
187             clr_IR<='0';
188             ld_IR<='0';
189             ld_PC<='0';
190             inc_PC<='0';
191             clr_A<='0';
192             ld_A<='1';
193             ld_B<='0';
194             clr_B<='0';
195             clr_C<='0';
196             ld_C<='0';
197             clr_Z<='0';
```

```
198             ld_Z<='0';
199             A_Mux<='0';
200             DATA_Mux<="01";
201             en<='1';
202             wen<='0';
203 
204             elsif Instruction_sig = "1010" then --LDB
205             clr_IR<='0';
206             ld_IR<='0';
207             ld_PC<='0';
208             inc_PC<='0';
209             clr_A<='0';
210             ld_A<='0';
211             ld_B<='1';
212             clr_B<='0';
213             clr_C<='0';
214             ld_C<='0';
215             clr_Z<='0';
216             ld_Z<='0';
217             B_Mux<='0';
218             DATA_Mux<="01";
219             en<='1';
220             wen<='1';
221 
222             elsif Instruction_sig = "0010" then --STA
223             clr_IR<='0';
224             ld_IR<='0';
225             ld_PC<='0';
226             inc_PC<='0';
227             clr_A<='0';
228             ld_A<='0';
229             ld_B<='0';
230             clr_B<='0';
231             clr_C<='0';
232             ld_C<='0';
233             clr_Z<='0';
234             ld_Z<='0';
235             REG_MUX<='0';
236             DATA_Mux<="00";
237             en<='1';
238             wen<='1';
239 
240             elsif Instruction_sig = "0011" then --STB
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             ld_B<='0';
248             clr_B<='0';
249             clr_C<='0';
250             ld_C<='0';
251             clr_Z<='0';
252             ld_Z<='0';
253             REG_MUX<='1';
254             DATA_Mux<="00";
255             en<='1';
256             wen<='1';
257 
258             elsif Instruction_sig = "0000" then --LDAI
259             clr_IR<='0';
260             ld_IR<='0';
261             ld_PC<='0';
262             inc_PC<='0';
263             clr_A<='0';
```

```
264             ld_A<='1';
265             ld_B<='0';
266             clr_B<='0';
267             clr_C<='0';
268             ld_C<='0';
269             clr_Z<='0';
270             ld_Z<='0';
271             A_Mux<='1';
272
273             elsif Instruction_sig = "0001" then --LDBI
274             clr_IR<='0';
275             ld_IR<='0';
276             ld_PC<='0';
277             inc_PC<='0';
278             clr_A<='0';
279             ld_A<='0';
280             ld_B<='1';
281             clr_B<='0';
282             clr_C<='0';
283             ld_C<='0';
284             clr_Z<='0';
285             ld_Z<='0';
286             B_Mux<='1';
287
288             elsif Instruction_sig = "0100" then --LUI
289             clr_IR<='0';
290             ld_IR<='0';
291             ld_PC<='0';
292             inc_PC<='0';
293             clr_A<='0';
294             ld_A<='1';
295             ld_B<='0';
296             clr_B<='1';
297             clr_C<='0';
298             ld_C<='0';
299             clr_Z<='0';
300             ld_Z<='0';
301             ALU_op<="001";
302             A_Mux<='0';
303             DATA_Mux<="10";
304             IM_MUX1<='1';
305
306             elsif Instruction_sig2 = "01111001" then --ANDI
307             clr_IR<='0';
308             ld_IR<='0';
309             ld_PC<='0';
310             inc_PC<='0';
311             clr_A<='0';
312             ld_A<='1';
313             ld_B<='0';
314             clr_B<='0';
315             clr_C<='0';
316             ld_C<='1';
317             clr_Z<='0';
318             ld_Z<='1';
319             ALU_op<="000";
320             A_Mux<='0';
321             DATA_Mux<="10";
322             IM_MUX1<='0';
323             IM_MUX2<="01";
324
325             elsif Instruction_sig2 = "01111110" then --DECA
326             clr_IR<='0';
327             ld_IR<='0';
328             ld_PC<='0';
329             inc_PC<='0';
```

```
330             clr_A<='0';
331             ld_A<='1';
332             ld_B<='0';
333             clr_B<='0';
334             clr_C<='0';
335             ld_C<='1';
336             clr_Z<='0';
337             ld_Z<='1';
338             ALU_op<="110";
339             A_Mux<='0';
340             DATA_Mux<="10";
341             IM_MUX1<='0';
342             IM_MUX2<="10";
343
344             elsif Instruction_sig2 = "01110000" then --ADD
345             clr_IR<='0';
346             ld_IR<='0';
347             ld_PC<='0';
348             inc_PC<='0';
349             clr_A<='0';
350             ld_A<='1';
351             ld_B<='0';
352             clr_B<='0';
353             clr_C<='0';
354             ld_C<='1';
355             clr_Z<='0';
356             ld_Z<='1';
357             ALU_op<="010";
358             A_Mux<='0';
359             DATA_Mux<="10";
360             IM_MUX1<='0';
361             IM_MUX2<="00";
362
363             elsif Instruction_sig2 = "01110010" then --SUB
364             clr_IR<='0';
365             ld_IR<='0';
366             ld_PC<='0';
367             inc_PC<='0';
368             clr_A<='0';
369             ld_A<='1';
370             ld_B<='0';
371             clr_B<='0';
372             clr_C<='0';
373             ld_C<='1';
374             clr_Z<='0';
375             ld_Z<='1';
376             ALU_op<="110";
377             A_Mux<='0';
378             DATA_Mux<="10";
379             IM_MUX1<='0';
380             IM_MUX2<="00";
381
382             elsif Instruction_sig2 = "01110011" then --INCA
383             clr_IR<='0';
384             ld_IR<='0';
385             ld_PC<='0';
386             inc_PC<='0';
387             clr_A<='0';
388             ld_A<='1';
389             ld_B<='0';
390             clr_B<='0';
391             clr_C<='0';
392             ld_C<='1';
393             clr_Z<='0';
394             ld_Z<='1';
395             ALU_op<="010";
```

```
396             A_Mux<='0';
397             DATA_Mux<="10";
398             IM_MUX1<='0';
399             IM_MUX2<="10";
400
401             elsif Instruction_sig2 = "01111011" then --AND
402             clr_IR<='0';
403             ld_IR<='0';
404             ld_PC<='0';
405             inc_PC<='0';
406             clr_A<='0';
407             ld_A<='1';
408             ld_B<='0';
409             clr_B<='0';
410             clr_C<='0';
411             ld_C<='1';
412             clr_Z<='0';
413             ld_Z<='1';
414             ALU_op<="000";
415             A_Mux<='0';
416             DATA_Mux<="10";
417             IM_MUX1<='0';
418             IM_MUX2<="00";
419
420
421             elsif Instruction_sig2 = "01110001" then --ADDI
422             clr_IR<='0';
423             ld_IR<='0';
424             ld_PC<='0';
425             inc_PC<='0';
426             clr_A<='0';
427             ld_A<='1';
428             ld_B<='0';
429             clr_B<='0';
430             clr_C<='0';
431             ld_C<='1';
432             clr_Z<='0';
433             ld_Z<='1';
434             ALU_op<="010";
435             A_Mux<='0';
436             DATA_Mux<="10";
437             IM_MUX1<='0';
438             IM_MUX2<="01";
439
440             elsif Instruction_sig2 = "01111101" then --ORI
441             clr_IR<='0';
442             ld_IR<='0';
443             ld_PC<='0';
444             inc_PC<='0';
445             clr_A<='0';
446             ld_A<='1';
447             ld_B<='0';
448             clr_B<='0';
449             clr_C<='0';
450             ld_C<='1';
451             clr_Z<='0';
452             ld_Z<='1';
453             ALU_op<="001";
454             A_Mux<='0';
455             DATA_Mux<="10";
456             IM_MUX1<='0';
457             IM_MUX2<="01";
458
459             elsif Instruction_sig2 = "01110100" then --ROL
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             ld_B<='0';
467             clr_B<='0';
468             clr_C<='0';
469             ld_C<='1';
470             clr_Z<='0';
471             ld_Z<='1';
472             ALU_op<="100";
473             A_Mux<='0';
474             DATA_Mux<="10";
475             IM_MUX1<='0';
476
477             elsif Instruction_sig2 = "01111111" then --ROR
478             clr_IR<='0';
479             ld_IR<='0';
480             ld_PC<='0';
481             inc_PC<='0';
482             clr_A<='0';
483             ld_A<='1';
484             ld_B<='0';
485             clr_B<='0';
486             clr_C<='0';
487             ld_C<='1';
488             clr_Z<='0';
489             ld_Z<='1';
490             ALU_op<="101";
491             A_Mux<='0';
492             DATA_Mux<="10";
493             IM_MUX1<='0';
494
495             elsif Instruction_sig2 = "01110101" then --CLRA
496             clr_IR<='0';
497             ld_IR<='0';
498             ld_PC<='0';
499             inc_PC<='0';
500             clr_A<='1';
501             ld_A<='0';
502             ld_B<='0';
503             clr_B<='0';
504             clr_C<='0';
505             ld_C<='0';
506             clr_Z<='0';
507             ld_Z<='0';
508
509             elsif Instruction_sig2 = "01110110" then --CLRB
510             clr_IR<='0';
511             ld_IR<='0';
512             ld_PC<='0';
513             inc_PC<='0';
514             clr_A<='0';
515             ld_A<='0';
516             ld_B<='0';
517             clr_B<='1';
518             clr_C<='0';
519             ld_C<='0';
520             clr_Z<='0';
521             ld_Z<='0';
522
523             elsif Instruction_sig2 = "01110111" then --CLRC
524             clr_IR<='0';
525             ld_IR<='0';
526             ld_PC<='0';
527             inc_PC<='0';
```

```
528             clr_A<='0';
529             ld_A<='0';
530             ld_B<='0';
531             clr_B<='0';
532             clr_C<='1';
533             ld_C<='0';
534             clr_Z<='0';
535             ld_Z<='0';
536 
537             elsif Instruction_sig2 = "01111000" then --CLRZ
538             clr_IR<='0';
539             ld_IR<='0';
540             ld_PC<='0';
541             inc_PC<='0';
542             clr_A<='0';
543             ld_A<='0';
544             ld_B<='0';
545             clr_B<='0';
546             clr_C<='0';
547             ld_C<='0';
548             clr_Z<='1';
549             ld_Z<='0';
550 
551             elsif Instruction_sig2 = "01111010" then --TSTZ
552                if(statusZ = '1') then
553             clr_IR<='0';
554             ld_IR<='0';
555             ld_PC<='1';
556             inc_PC<='1';
557             clr_A<='0';
558             ld_A<='0';
559             ld_B<='0';
560             clr_B<='0';
561             clr_C<='0';
562             ld_C<='0';
563             clr_Z<='0';
564             ld_Z<='0';
565             end if;
566 
567             elsif Instruction_sig2 = "01111100" then --TSTC
568             if(statusC = '1') then
569             clr_IR<='0';
570             ld_IR<='0';
571             ld_PC<='1';
572             inc_PC<='1';
573             clr_A<='0';
574             ld_A<='0';
575             ld_B<='0';
576             clr_B<='0';
577             clr_C<='0';
578             ld_C<='0';
579             clr_Z<='0';
580             ld_Z<='0';
581             end if;
582             end if;
583             end if;
584             end if;
585             end process;
586 
587    -----STATE MACHINE-----
588    process (clk, enable)
589    begin
590       if enable = '1' then
591       if rising_edge (clk) then
592       if present_state = state_0 then present_state <= state_1;
593       elsif present_state = state_1 then present_state <= state_2;
```

```
594       else present_state <= state_0;
595       end if;
596       end if;
597       else present_state <= state_0;
598       end if;
599       end process;
600
601       with present_state select
602       T <= "001" when state_0,
603            "010" when state_1,
604            "100" when state_2,
605            "001" when others;
606   end description;
607
```