```
1   library ieee;
2   use ieee.std_logic_1164.ALL;
3   ENTITY control IS
4   PORT(
5   clk, mclk : IN STD_LOGIC;
6   enable : IN STD_LOGIC;
7   statusC, statusZ : IN STD_LOGIC;
8   INST : IN STD_LOGIC_VECTOR(31 DOWNTO 0);
9   PC_Mux : OUT STD_LOGIC;
10  IM_MUX1, REG_Mux : OUT STD_LOGIC;
11  IM_MUX2, DATA_Mux : OUT STD_LOGIC_VECTOR(1 DOWNTO 0);
12  ALU_op : OUT STD_LOGIC_VECTOR(2 DOWNTO 0);
13  inc_PC, ld_PC : OUT STD_LOGIC;
14  clr_IR : OUT STD_LOGIC;
15  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  END control;
21
22  ARCHITECTURE description OF control IS
23  TYPE STATETYPE IS (state_0, state_1, state_2);
24  SIGNAL present_state: STATETYPE;
25
26  BEGIN
27  -------- OPERATION DECODER ---------
28  PROCESS (present_state, INST, statusC, statusZ, enable)
29  BEGIN
30  case enable is
31  when '0' =>
32     PC_Mux <= '0';
33     IM_MUX1 <= '0';
34     REG_Mux <= '0';
35     IM_MUX2 <= "00";
36     DATA_Mux <= "00";
37     ALU_op <= "000";
38     inc_PC <= '0';
39     ld_PC <= '0';
40     clr_IR <= '0';
41     ld_IR <= '0';
42     clr_A <= '0';
43     clr_B <= '0';
44     clr_C <= '0';
45     clr_Z <= '0';
46     ld_A <= '0';
47     ld_B <= '0';
48     ld_C <= '0';
49     ld_Z <= '0';
50  when '1' =>
51     case present_state is
52     when state_0 =>
53        PC_Mux <= '0';
54        IM_MUX1 <= '0';
55        REG_Mux <= '0';
56        IM_MUX2 <= "00";
57        DATA_Mux <= "00";
58        ALU_op <= "000";
59        inc_PC <= '0';
60        ld_PC <= '0';
61        clr_IR <= '0';
62        ld_IR <= '1';
```

```
63          clr_A <= '0';
64          clr_B <= '0';
65          clr_C <= '0';
66          clr_Z <= '0';
67          ld_A <= '0';
68          ld_B <= '0';
69          ld_C <= '0';
70          ld_Z <= '0';
71       when state_1 =>
72          ld_PC <= '1';
73          inc_PC <= '1';
74          if(INST(31 downto 28) = "0000") then -- LDAI
75             PC_Mux <= '0';
76             IM_MUX1 <= '0';
77             REG_Mux <= '0';
78             IM_MUX2 <= "00";
79             DATA_Mux <= "00";
80             ALU_op <= "000";
81             clr_IR <= '0';
82             ld_IR <= '0';
83             clr_A <= '0';
84             clr_B <= '0';
85             clr_C <= '0';
86             clr_Z <= '0';
87             ld_A <= '1';
88             ld_B <= '0';
89             ld_C <= '0';
90             ld_Z <= '0';
91          elsif(INST(31 downto 28) = "0001") then -- LDBI
92             PC_Mux <= '0';
93             IM_MUX1 <= '0';
94             REG_Mux <= '0';
95             IM_MUX2 <= "00";
96             DATA_Mux <= "00";
97             ALU_op <= "000";
98             clr_IR <= '0';
99             ld_IR <= '0';
100            clr_A <= '0';
101            clr_B <= '0';
102            clr_C <= '0';
103            clr_Z <= '0';
104            ld_A <= '0';
105            ld_B <= '1';
106            ld_C <= '0';
107            ld_Z <= '0';
108         elsif(INST(31 downto 28) = "0010") then -- STA
109            PC_Mux <= '0';
110            IM_MUX1 <= '0';
111            REG_Mux <= '0';
112            IM_MUX2 <= "00";
113            DATA_Mux <= "00";
114            ALU_op <= "000";
115            clr_IR <= '0';
116            ld_IR <= '0';
117            clr_A <= '0';
118            clr_B <= '0';
119            clr_C <= '0';
120            clr_Z <= '0';
121            ld_A <= '0';
122            ld_B <= '0';
123            ld_C <= '0';
124            ld_Z <= '0';
```

```
125         elsif(INST(31 downto 28) = "0011") then -- STB
126            PC_Mux <= '0';
127            IM_MUX1 <= '0';
128            REG_Mux <= '1';
129            IM_MUX2 <= "00";
130            DATA_Mux <= "00";
131            ALU_op <= "000";
132            clr_IR <= '0';
133            ld_IR <= '0';
134            clr_A <= '0';
135            clr_B <= '0';
136            clr_C <= '0';
137            clr_Z <= '0';
138            ld_A <= '0';
139            ld_B <= '0';
140            ld_C <= '0';
141            ld_Z <= '0';
142         elsif(INST(31 downto 28) = "1001") then -- LDA
143            PC_Mux <= '0';
144            IM_MUX1 <= '0';
145            REG_Mux <= '0';
146            IM_MUX2 <= "00";
147            DATA_Mux <= "01";
148            ALU_op <= "000";
149            clr_IR <= '0';
150            ld_IR <= '0';
151            clr_A <= '0';
152            clr_B <= '0';
153            clr_C <= '0';
154            clr_Z <= '0';
155            ld_A <= '1';
156            ld_B <= '0';
157            ld_C <= '0';
158            ld_Z <= '0';
159         elsif(INST(31 downto 28) = "1010") then -- LDB
160            PC_Mux <= '0';
161            IM_MUX1 <= '0';
162            REG_Mux <= '1';
163            IM_MUX2 <= "00";
164            DATA_Mux <= "01";
165            ALU_op <= "000";
166            clr_IR <= '0';
167            ld_IR <= '0';
168            clr_A <= '0';
169            clr_B <= '0';
170            clr_C <= '0';
171            clr_Z <= '0';
172            ld_A <= '0';
173            ld_B <= '1';
174            ld_C <= '0';
175            ld_Z <= '0';
176         else
177            PC_Mux <= '0';
178            IM_MUX1 <= '0';
179            REG_Mux <= '0';
180            IM_MUX2 <= "00";
181            DATA_Mux <= "00";
182            ALU_op <= "000";
183            clr_IR <= '0';
184            ld_IR <= '1';
185            clr_A <= '0';
186            clr_B <= '0';
```

```
187             clr_C <= '0';
188             clr_Z <= '0';
189             ld_A <= '0';
190             ld_B <= '0';
191             ld_C <= '0';
192             ld_Z <= '0';
193          end if;
194       when state_2 =>
195          ld_PC <= '0';
196          inc_PC <= '0';
197          if(INST(31 downto 28) = "0100") then -- LUI
198             PC_Mux <= '0';
199             IM_MUX1 <= '1';
200             REG_Mux <= '0';
201             IM_MUX2 <= "00";
202             DATA_Mux <= "10";
203             ALU_op <= "010";
204             clr_IR <= '0';
205             ld_IR <= '1';
206             clr_A <= '0';
207             clr_B <= '1';
208             clr_C <= '0';
209             clr_Z <= '0';
210             ld_A <= '1';
211             ld_B <= '0';
212             ld_C <= '1';
213             ld_Z <= '1';
214          elsif(INST(31 downto 28) = "0101") then -- JMP
215             PC_Mux <= '0';
216             IM_MUX1 <= '0';
217             REG_Mux <= '0';
218             IM_MUX2 <= "00";
219             DATA_Mux <= "00";
220             ALU_op <= "000";
221             clr_IR <= '0';
222             ld_IR <= '1';
223             clr_A <= '0';
224             clr_B <= '0';
225             clr_C <= '0';
226             clr_Z <= '0';
227             ld_A <= '0';
228             ld_B <= '0';
229             ld_C <= '0';
230             ld_Z <= '0';
231          elsif(INST(31 downto 28) = "0110") then -- BEQ
232             if(statusZ = '1') then
233                PC_Mux <= '0';
234                IM_MUX1 <= '0';
235                REG_Mux <= '0';
236                IM_MUX2 <= "00";
237                DATA_Mux <= "00";
238                ALU_op <= "011";
239                inc_PC <= '0';
240                ld_PC <= '1';
241                clr_IR <= '0';
242                ld_IR <= '1';
243                clr_A <= '0';
244                clr_B <= '0';
245                clr_C <= '0';
246                clr_Z <= '0';
247                ld_A <= '0';
248                ld_B <= '0';
```

```
249               ld_C <= '0';
250               ld_Z <= '0';
251            end if;
252         elsif(INST(31 downto 28) = "1000") then -- BCO
253            if(statusC = '1') then
254               PC_Mux <= '0';
255               IM_MUX1 <= '0';
256               REG_Mux <= '0';
257               IM_MUX2 <= "00";
258               DATA_Mux <= "00";
259               ALU_op <= "000";
260               inc_PC <= '0';
261               ld_PC <= '1';
262               clr_IR <= '0';
263               ld_IR <= '0';
264               clr_A <= '0';
265               clr_B <= '0';
266               clr_C <= '0';
267               clr_Z <= '0';
268               ld_A <= '0';
269               ld_B <= '0';
270               ld_C <= '0';
271               ld_Z <= '0';
272            end if;
273         elsif(INST(31 downto 28) = "0111") then
274            CASE INST(27 downto 24) is
275               when "0000" => -- ADD
276                  PC_Mux <= '0';
277                  IM_MUX1 <= '0';
278                  REG_Mux <= '0';
279                  IM_MUX2 <= "00";
280                  DATA_Mux <= "10";
281                  ALU_op <= "010";
282                  clr_IR <= '0';
283                  ld_IR <= '1';
284                  clr_A <= '0';
285                  clr_B <= '0';
286                  clr_C <= '0';
287                  clr_Z <= '0';
288                  ld_A <= '1';
289                  ld_B <= '0';
290                  ld_C <= '1';
291                  ld_Z <= '1';
292               when "0001" => -- ADDI
293                  PC_Mux <= '0';
294                  IM_MUX1 <= '0';
295                  REG_Mux <= '0';
296                  IM_MUX2 <= "01";
297                  DATA_Mux <= "10";
298                  ALU_op <= "010";
299                  clr_IR <= '0';
300                  ld_IR <= '1';
301                  clr_A <= '0';
302                  clr_B <= '0';
303                  clr_C <= '0';
304                  clr_Z <= '0';
305                  ld_A <= '1';
306                  ld_B <= '0';
307                  ld_C <= '1';
308                  ld_Z <= '1';
309               when "0010" => -- SUB
310                  PC_Mux <= '0';
```

```
311                 IM_MUX1 <= '0';
312                 REG_Mux <= '0';
313                 IM_MUX2 <= "00";
314                 DATA_Mux <= "10";
315                 ALU_op <= "110";
316                 clr_IR <= '0';
317                 ld_IR <= '1';
318                 clr_A <= '0';
319                 clr_B <= '0';
320                 clr_C <= '0';
321                 clr_Z <= '0';
322                 ld_A <= '1';
323                 ld_B <= '0';
324                 ld_C <= '1';
325                 ld_Z <= '1';
326             when "0011" => -- INCA
327                 PC_Mux <= '0';
328                 IM_MUX1 <= '0';
329                 REG_Mux <= '0';
330                 IM_MUX2 <= "10";
331                 DATA_Mux <= "10";
332                 ALU_op <= "010";
333                 clr_IR <= '0';
334                 ld_IR <= '1';
335                 clr_A <= '0';
336                 clr_B <= '0';
337                 clr_C <= '0';
338                 clr_Z <= '0';
339                 ld_A <= '1';
340                 ld_B <= '0';
341                 ld_C <= '1';
342                 ld_Z <= '1';
343             when "0100" => -- MUL2B
344                 PC_Mux <= '0';
345                 IM_MUX1 <= '0';
346                 REG_Mux <= '0';
347                 IM_MUX2 <= "00";
348                 DATA_Mux <= "10";
349                 ALU_op <= "100";
350                 clr_IR <= '0';
351                 ld_IR <= '1';
352                 clr_A <= '0';
353                 clr_B <= '0';
354                 clr_C <= '0';
355                 clr_Z <= '0';
356                 ld_A <= '1';
357                 ld_B <= '0';
358                 ld_C <= '1';
359                 ld_Z <= '1';
360             when "0101" => -- CLRA
361                 PC_Mux <= '0';
362                 IM_MUX1 <= '0';
363                 REG_Mux <= '0';
364                 IM_MUX2 <= "00";
365                 DATA_Mux <= "00";
366                 ALU_op <= "000";
367                 clr_IR <= '0';
368                 ld_IR <= '1';
369                 clr_A <= '1';
370                 clr_B <= '0';
371                 clr_C <= '0';
372                 clr_Z <= '0';
```

```
373                 ld_A <= '0';
374                 ld_B <= '0';
375                 ld_C <= '0';
376                 ld_Z <= '0';
377             when "0110" => -- CLRB
378                 PC_Mux <= '0';
379                 IM_MUX1 <= '0';
380                 REG_Mux <= '0';
381                 IM_MUX2 <= "00";
382                 DATA_Mux <= "00";
383                 ALU_op <= "000";
384                 clr_IR <= '0';
385                 ld_IR <= '1';
386                 clr_A <= '0';
387                 clr_B <= '1';
388                 clr_C <= '0';
389                 clr_Z <= '0';
390                 ld_A <= '0';
391                 ld_B <= '0';
392                 ld_C <= '0';
393                 ld_Z <= '0';
394             when "0111" => -- CLRC
395                 PC_Mux <= '0';
396                 IM_MUX1 <= '0';
397                 REG_Mux <= '0';
398                 IM_MUX2 <= "00";
399                 DATA_Mux <= "00";
400                 ALU_op <= "000";
401                 clr_IR <= '0';
402                 ld_IR <= '1';
403                 clr_A <= '0';
404                 clr_B <= '0';
405                 clr_C <= '1';
406                 clr_Z <= '0';
407                 ld_A <= '0';
408                 ld_B <= '0';
409             when "1000" => -- CLRZ
410                 PC_Mux <= '0';
411                 IM_MUX1 <= '0';
412                 REG_Mux <= '0';
413                 IM_MUX2 <= "00";
414                 DATA_Mux <= "00";
415                 ALU_op <= "000";
416                 clr_IR <= '0';
417                 ld_IR <= '1';
418                 clr_A <= '0';
419                 clr_B <= '0';
420                 clr_C <= '0';
421                 clr_Z <= '1';
422                 ld_A <= '0';
423                 ld_B <= '0';
424                 ld_C <= '0';
425                 ld_Z <= '0';
426             when "1001" => -- SCO
427                 PC_Mux <= '0';
428                 IM_MUX1 <= '0';
429                 REG_Mux <= '0';
430                 IM_MUX2 <= "00";
431                 DATA_Mux <= "00";
432                 ALU_op <= "111";
433                 clr_IR <= '0';
434                 ld_IR <= '1';
```

```
435                     clr_A <= '0';
436                     clr_B <= '0';
437                     clr_C <= '0';
438                     clr_Z <= '0';
439                     ld_A <= '0';
440                     ld_B <= '0';
441                     ld_C <= '1';
442                     ld_Z <= '0';
443                 when "1010" => -- TSTZ
444                     if(statusZ = '1') then
445                         PC_Mux <= '1';
446                         IM_MUX1 <= '0';
447                         REG_Mux <= '0';
448                         IM_MUX2 <= "00";
449                         DATA_Mux <= "00";
450                         ALU_op <= "000";
451                         inc_PC <= '1';
452                         ld_PC <= '1';
453                         clr_IR <= '0';
454                         ld_IR <= '1';
455                         clr_A <= '0';
456                         clr_B <= '0';
457                         clr_C <= '0';
458                         clr_Z <= '0';
459                         ld_A <= '0';
460                         ld_B <= '0';
461                         ld_C <= '0';
462                         ld_Z <= '0';
463                     end if;
464                 when "1011" => -- AND
465                     PC_Mux <= '0';
466                     IM_MUX1 <= '0';
467                     REG_Mux <= '0';
468                     IM_MUX2 <= "00";
469                     DATA_Mux <= "10";
470                     ALU_op <= "000";
471                     clr_IR <= '0';
472                     ld_IR <= '1';
473                     clr_A <= '0';
474                     clr_B <= '0';
475                     clr_C <= '0';
476                     clr_Z <= '0';
477                     ld_A <= '1';
478                     ld_B <= '0';
479                     ld_C <= '1';
480                     ld_Z <= '1';
481                 when "1100" => -- SEQ
482                     PC_Mux <= '0';
483                     IM_MUX1 <= '0';
484                     REG_Mux <= '0';
485                     IM_MUX2 <= "00";
486                     DATA_Mux <= "00";
487                     ALU_op <= "011";
488                     clr_IR <= '0';
489                     ld_IR <= '1';
490                     clr_A <= '0';
491                     clr_B <= '0';
492                     clr_C <= '0';
493                     clr_Z <= '0';
494                     ld_A <= '0';
495                     ld_B <= '0';
496                     ld_C <= '1';
```

```
497                 ld_Z <= '1';
498             when "1101" => -- ORI
499                 PC_Mux <= '0';
500                 IM_MUX1 <= '0';
501                 REG_Mux <= '0';
502                 IM_MUX2 <= "01";
503                 DATA_Mux <= "10";
504                 ALU_op <= "001";
505                 clr_IR <= '0';
506                 ld_IR <= '0';
507                 clr_A <= '0';
508                 clr_B <= '0';
509                 clr_C <= '0';
510                 clr_Z <= '0';
511                 ld_A <= '1';
512                 ld_B <= '0';
513                 ld_C <= '1';
514                 ld_Z <= '1';
515             when "1110" => -- DECA
516                 PC_Mux <= '0';
517                 IM_MUX1 <= '0';
518                 REG_Mux <= '0';
519                 IM_MUX2 <= "10";
520                 DATA_Mux <= "10";
521                 ALU_op <= "110";
522                 clr_IR <= '0';
523                 ld_IR <= '1';
524                 clr_A <= '0';
525                 clr_B <= '0';
526                 clr_C <= '0';
527                 clr_Z <= '0';
528                 ld_A <= '1';
529                 ld_B <= '0';
530                 ld_C <= '1';
531                 ld_Z <= '1';
532             when "1111" => -- DIV2B
533                 PC_Mux <= '0';
534                 IM_MUX1 <= '0';
535                 REG_Mux <= '0';
536                 IM_MUX2 <= "00";
537                 DATA_Mux <= "10";
538                 ALU_op <= "101";
539                 clr_IR <= '0';
540                 ld_IR <= '1';
541                 clr_A <= '0';
542                 clr_B <= '0';
543                 clr_C <= '0';
544                 clr_Z <= '0';
545                 ld_A <= '1';
546                 ld_B <= '0';
547                 ld_C <= '1';
548                 ld_Z <= '1';
549             when others =>
550                 PC_Mux <= '0';
551                 IM_MUX1 <= '0';
552                 REG_Mux <= '0';
553                 IM_MUX2 <= "00";
554                 DATA_Mux <= "00";
555                 ALU_op <= "000";
556                 clr_IR <= '0';
557                 ld_IR <= '0';
558                 clr_A <= '0';
```

```
559                  clr_B <= '0';
560                  clr_C <= '0';
561                  clr_Z <= '0';
562                  ld_A <= '0';
563                  ld_B <= '0';
564                  ld_C <= '0';
565                  ld_Z <= '0';
566            end case;
567         end if;
568      when others =>
569         PC_Mux <= '0';
570         IM_MUX1 <= '0';
571         REG_Mux <= '0';
572         IM_MUX2 <= "00";
573         DATA_Mux <= "00";
574         ALU_op <= "000";
575         ld_PC <= '0';
576         inc_PC <= '0';
577         clr_IR <= '0';
578         ld_IR <= '1';
579         clr_A <= '0';
580         clr_B <= '0';
581         clr_C <= '0';
582         clr_Z <= '0';
583         ld_A <= '0';
584         ld_B <= '0';
585         ld_C <= '0';
586         ld_Z <= '0';
587      end case;
588   end case;
589   END process;
590
591
592
593   -------- STATE MACHINE ---------
594   PROCESS (clk, enable)begin
595   case enable is
596   when '1' =>
597      if (clk'event and clk = '1') then
598         case present_state is
599            when state_0 =>
600               present_state <= state_1;
601               T <= "010";
602            when state_1 =>
603               present_state <= state_2;
604               T <= "100";
605            when state_2 =>
606               present_state <= state_0;
607               T <= "001";
608         end case;
609      end if;
610   when others =>
611      present_state <= state_0;
612   end case;
613   END process;
614
615
616
617   -------- DATA MEMORY INSTRUCTIONS ---------
618   PROCESS (mclk, clk, INST)
619   BEGIN
620
```

```
621   IF(mclk'EVENT and mclk = '0') THEN
622   IF(present_state = state_1 AND clk = '0') THEN
623      if(INST(31 downto 28) = "0000") then -- LDAI
624         en <= '0';
625         wen <= '0';
626      elsif(INST(31 downto 28) = "0001") then -- LDBI
627         en <= '0';
628         wen <= '0';
629      elsif(INST(31 downto 28) = "0010") then -- STA
630         en <= '1';
631         wen <= '1';
632      elsif(INST(31 downto 28) = "0011") then -- STB
633         en <= '1';
634         wen <= '1';
635      elsif(INST(31 downto 28) = "1001") then -- LDA
636         en <= '1';
637         wen <= '0';
638      elsif(INST(31 downto 28) = "1010") then -- LDB
639         en <= '1';
640         wen <= '0';
641      else
642         en <= '0';
643         wen <= '0';
644      end if;
645   ELSIF(present_state = state_2 AND clk = '1') THEN
646      if(INST(31 downto 28) = "0000") then -- LDAI
647         en <= '0';
648         wen <= '0';
649      elsif(INST(31 downto 28) = "0001") then -- LDBI
650         en <= '0';
651         wen <= '0';
652      elsif(INST(31 downto 28) = "0010") then -- STA
653         en <= '0';
654         wen <= '0';
655      elsif(INST(31 downto 28) = "0011") then -- STB
656         en <= '0';
657         wen <= '0';
658      elsif(INST(31 downto 28) = "1001") then -- LDA
659         en <= '0';
660         wen <= '0';
661      elsif(INST(31 downto 28) = "1010") then -- LDB
662         en <= '0';
663         wen <= '0';
664      else
665         en <= '0';
666         wen <= '0';
667      end if;
668   ELSIF(present_state = state_1) THEN
669      if(INST(31 downto 28) = "0000") then -- LDAI
670         en <= '0';
671         wen <= '0';
672      elsif(INST(31 downto 28) = "0001") then -- LDBI
673         en <= '0';
674         wen <= '0';
675      elsif(INST(31 downto 28) = "0010") then -- STA
676         en <= '1';
677         wen <= '1';
678      elsif(INST(31 downto 28) = "0011") then -- STB
679         en <= '1';
680         wen <= '1';
681      elsif(INST(31 downto 28) = "1001") then -- LDA
682         en <= '1';
```

```
683         wen <= '0';
684      elsif(INST(31 downto 28) = "1010") then -- LDB
685         en <= '1';
686         wen <= '0';
687      else
688         en <= '0';
689         wen <= '0';
690      end if;
691
692   END IF;
693   END IF;
694   END process;
695   END description;
```