

```

1 module CONTROL_UNIT(
2     input clk,
3     input [0:0] z,
4     input [0:0] i,
5     input [0:0] j,
6     input [0:0] k,
7     input [7:0] instruction,
8     input [15:0]MDDR,
9
10    output reg[3:0] to_ALU,
11    output reg[0:0] to_DM,
12    output reg[0:0] to_IM,
13    output reg[5:0] to_Decoder,
14    output reg[4:0] to_BUS,
15    output reg[0:0] to_PC,
16    output reg[2:0] to_AC,
17    output reg[0:0] en_RegSelector,
18    output reg[0:0] rw_RegSelector,
19    output reg finish,
20    output reg [7:0]current_micro_instruction);
21
22    reg [5:0] CS = 8'd3;
23    reg [5:0] NS = 8'd0;
24
25    parameter FETCH1 = 8'd0;
26    parameter FETCH2 = 8'd1;
27    parameter FETCH3 = 8'd2;
28    parameter NOP = 8'd3;
29    parameter END = 8'd4;
30    parameter CLR1 = 8'd5;
31    parameter CLR2 = 8'd6;
32    parameter CLR3 = 8'd7;
33    parameter LOAD1 = 8'd8;
34    parameter LOAD2 = 8'd9;
35    parameter LOAD3 = 8'd10;
36    parameter LOAD4 = 8'd11;
37    parameter LOAD5 = 8'd12;
38    parameter LOAD6 = 8'd13;
39    parameter LOADM1 = 8'd14;
40    parameter LOADM2 = 8'd15;
41    parameter LOADM3 = 8'd16;
42    parameter LOADM4 = 8'd17;
43    parameter STAC1 = 8'd18;
44    parameter STAC2 = 8'd19;
45    parameter STAC3 = 8'd20;
46    parameter INC1 = 8'd21;
47    parameter INC2 = 8'd22;
48    parameter INC3 = 8'd23;
49    parameter INC4 = 8'd24;
50    parameter INC5 = 8'd25;
51    parameter INC6 = 8'd26;
52    parameter INCAC1 = 8'd27;
53    parameter JUMP1 = 8'd28;
54    parameter JUMP2 = 8'd29;

```

```

55     parameter JUMPY1 = 8'd30;
56     parameter JUMPY2 = 8'd31;
57     parameter JUMPY3 = 8'd32;
58     parameter JUMPN1 = 8'd33;
59     parameter MOVE1 = 8'd34;
60     parameter MOVE2 = 8'd35;
61     parameter MOVE3 = 8'd36;
62     parameter MOVE4 = 8'd37;
63     parameter MOVE5 = 8'd38;
64     parameter ADD = 8'd39;
65     parameter SUB = 8'd40;
66     parameter MUL = 8'd41;
67     parameter AND = 8'd42;
68     parameter OR = 8'd43;
69     parameter STORM1 = 8'd44;
70     parameter STORM2 = 8'd45;
71
72     always @(posedge clk)
73     begin
74         CS <= NS;
75         current_micro_instruction<=NS;
76     end
77     always @(posedge clk)
78     begin
79         if (CS == END)
80             finish <= 1'd1;
81         else
82             finish <= 1'd0;
83     end
84     always @(CS or z or instruction)
85     begin
86         case(CS)
87             NOP: begin
88                 NS = FETCH1;
89             end
90             FETCH1: begin
91                 to_ALU = 4'd0;
92                 to_BUS = 5'd2;
93                 to_Decoder = 5'd1;
94                 to_IM = 1'd0;
95                 to_DM = 1'd0;
96                 to_PC = 1'd1;
97                 to_AC = 3'd0;
98                 en_RegSelector = 1'd0;
99                 rw_RegSelector = 1'd0;
100                NS = FETCH2;
101            end
102            FETCH2: begin
103                to_ALU = 4'd0;
104                to_BUS = 5'd2;
105                to_Decoder = 5'd18;
106                to_IM = 1'd0;
107                to_DM = 1'd0;
108                to_PC = 1'd0;

```

```

109         to_AC = 3'd0;
110         en_RegSelector = 1'd0;
111         rw_RegSelector = 1'd0;
112         NS = FETCH3;
113     end
114     FETCH3: begin
115         to_ALU = 4'd0;
116         to_BUS = 5'd4;
117         to_Decoder = 5'd3;
118         to_IM = 1'd0;
119         to_DM = 1'd0;
120         to_PC = 1'd0;
121         to_AC = 3'd0;
122         en_RegSelector = 1'd0;
123         rw_RegSelector = 1'd0;
124         NS = instruction[7:0];
125     end
126     END: begin
127         NS = END;
128     end
129     CLR1: begin
130         to_ALU = 4'd0;
131         to_BUS = 5'd4;
132         to_Decoder = 5'd6;
133         to_IM = 1'd0;
134         to_DM = 1'd0;
135         to_PC = 1'd1;
136         to_AC = 3'd0;
137         en_RegSelector = 1'd0;
138         rw_RegSelector = 1'd0;
139         NS = CLR2;
140     end
141     CLR2: begin
142         to_ALU = 4'd8;
143         to_BUS = 5'd0;
144         to_Decoder = 5'd0;
145         to_IM = 1'd0;
146         to_DM = 1'd0;
147         to_PC = 1'd0;
148         to_AC = 3'b000;
149         en_RegSelector = 1'd0;
150         rw_RegSelector = 1'd0;
151         NS = CLR3;
152     end
153     CLR3: begin
154         to_ALU = 4'd0;
155         to_BUS = 5'd0;
156         to_Decoder = 5'd0;
157         to_IM = 1'd0;
158         to_DM = 1'd0;
159         to_PC = 1'd0;
160         to_AC = 3'b000;
161         en_RegSelector = 1'd1;
162         rw_RegSelector = 1'd1;

```

```

163         NS = FETCH1;
164     end
165     LOAD1: begin
166         to_ALU = 4'd0;
167         to_BUS = 5'd0;
168         to_Decoder = 5'd6;
169         to_IM = 1'd0;
170         to_DM = 1'd0;
171         to_PC = 1'd1;
172         to_AC = 3'b000;
173         en_RegSelector = 1'd0;
174         rw_RegSelector = 1'd0;
175         NS = LOAD2;
176     end
177     LOAD2: begin
178         to_ALU = 4'd0;
179         to_BUS = 5'd4;
180         to_Decoder = 5'd1;
181         to_IM = 1'd0;
182         to_DM = 1'd0;
183         to_PC = 1'd0;
184         to_AC = 3'b000;
185         en_RegSelector = 1'd0;
186         rw_RegSelector = 1'd0;
187         NS = LOAD3;
188     end
189     LOAD3: begin
190         to_ALU = 4'd0;
191         to_BUS = 5'd2;
192         to_Decoder = 5'd19;
193         to_IM = 1'd0;
194         to_DM = 1'd0;
195         to_PC = 1'd0;
196         to_AC = 3'b000;
197         en_RegSelector = 1'd0;
198         rw_RegSelector = 1'd0;
199         NS = LOAD4;
200     end
201     LOAD4: begin
202         to_ALU = 4'd1;
203         to_BUS = 5'd4;
204         to_Decoder = 5'd0;
205         to_IM = 1'd0;
206         to_DM = 1'd0;
207         to_PC = 1'd0;
208         to_AC = 3'b100;
209         en_RegSelector = 1'd0;
210         rw_RegSelector = 1'd0;
211         NS = LOAD5;
212     end
213     LOAD5: begin
214         to_ALU = 4'd0;
215         to_BUS = 5'd0;
216         to_Decoder = 5'd6;

```

```

217         to_IM = 1'd0;
218         to_DM = 1'd0;
219         to_PC = 1'd1;
220         to_AC = 3'b000;
221         en_RegSelector = 1'd0;
222         rw_RegSelector = 1'd0;
223         NS = LOAD6;
224     end
225     LOAD6: begin
226         to_ALU = 4'd0;
227         to_BUS = 5'd20;
228         to_Decoder = 5'd0;
229         to_IM = 1'd0;
230         to_DM = 1'd0;
231         to_PC = 1'd0;
232         to_AC = 3'b000;
233         en_RegSelector = 1'd1;
234         rw_RegSelector = 1'd1;
235         NS = FETCH1;
236     end
237     LOADM1: begin
238         to_ALU = 4'd0;
239         to_BUS = 5'd0;
240         to_Decoder = 5'd6;
241         to_IM = 1'd0;
242         to_DM = 1'd0;
243         to_PC = 1'd1;
244         to_AC = 3'b000;
245         en_RegSelector = 1'd0;
246         rw_RegSelector = 1'd0;
247         NS = LOADM2;
248     end
249     LOADM2: begin
250         to_ALU = 4'd0;
251         to_BUS = 5'd0;
252         to_Decoder = 5'd1;
253         to_IM = 1'd0;
254         to_DM = 1'd0;
255         to_PC = 1'd0;
256         to_AC = 3'b000;
257         en_RegSelector = 1'd1;
258         rw_RegSelector = 1'd0;
259         NS = LOADM3;
260     end
261     LOADM3: begin
262         to_ALU = 4'd0;
263         to_BUS = 5'd0;
264         to_Decoder = 5'd5;
265         to_IM = 1'd0;
266         to_DM = 1'd0;
267         to_PC = 1'd0;
268         to_AC = 3'b000;
269         en_RegSelector = 1'd0;
270         rw_RegSelector = 1'd0;

```

```

271         NS = LOADM4;
272     end
273     LOADM4: begin
274         to_ALU = 4'd1;
275         to_BUS = 5'd4;
276         to_Decoder = 5'd0;
277         to_IM = 1'd0;
278         to_DM = 1'd0;
279         to_PC = 1'd0;
280         to_AC = 3'b100;
281         en_RegSelector = 1'd0;
282         rw_RegSelector = 1'd0;
283         NS = FETCH1;
284     end
285     STAC1: begin
286         to_ALU = 4'd0;
287         to_BUS = 5'd0;
288         to_Decoder = 5'd6;
289         to_IM = 1'd0;
290         to_DM = 1'd0;
291         to_PC = 1'd1;
292         to_AC = 3'b000;
293         en_RegSelector = 1'd0;
294         rw_RegSelector = 1'd0;
295         NS = STAC2;
296     end
297     STAC2: begin
298         to_ALU = 4'd0;
299         to_BUS = 5'd4;
300         to_Decoder = 5'd1;
301         to_IM = 1'd0;
302         to_DM = 1'd0;
303         to_PC = 1'd0;
304         to_AC = 3'b000;
305         en_RegSelector = 1'd0;
306         rw_RegSelector = 1'd0;
307         NS = STAC3;
308     end
309     STAC3: begin
310         to_ALU = 4'd0;
311         to_BUS = 5'd20;
312         to_Decoder = 5'd4;
313         to_IM = 1'd0;
314         to_DM = 1'd1;
315         to_PC = 1'd0;
316         to_AC = 3'b000;
317         en_RegSelector = 1'd0;
318         rw_RegSelector = 1'd0;
319         NS = FETCH1;
320     end
321     INC1: begin
322         to_ALU = 4'd0;
323         to_BUS = 5'd0;
324         to_Decoder = 5'd6;

```

```

325         to_IM = 1'd0;
326         to_DM = 1'd0;
327         to_PC = 1'd1;
328         to_AC = 3'b000;
329         en_RegSelector = 1'd0;
330         rw_RegSelector = 1'd0;
331         NS = INC2;
332     end
333     INC2: begin
334         to_ALU = 4'd1;
335         to_BUS = 5'd4;
336         to_Decoder = 5'd0;
337         to_IM = 1'd0;
338         to_DM = 1'd0;
339         to_PC = 1'd0;
340         to_AC = 3'b100;
341         en_RegSelector = 1'd0;
342         rw_RegSelector = 1'd0;
343         NS = INC3;
344     end
345     INC3: begin
346         to_ALU = 4'd0;
347         to_BUS = 5'd2;
348         to_Decoder = 5'd1;
349         to_IM = 1'd0;
350         to_DM = 1'd0;
351         to_PC = 1'd0;
352         to_AC = 3'b000;
353         en_RegSelector = 1'd0;
354         rw_RegSelector = 1'd0;
355         NS = INC4;
356     end
357     INC4: begin
358         to_ALU = 4'd0;
359         to_BUS = 5'd0;
360         to_Decoder = 5'd6;
361         to_IM = 1'd0;
362         to_DM = 1'd0;
363         to_PC = 1'd1;
364         to_AC = 3'b000;
365         en_RegSelector = 1'd0;
366         rw_RegSelector = 1'd0;
367         NS = INC5;
368     end
369     INC5: begin
370         to_ALU = 4'd2;
371         to_BUS = 5'd0;
372         to_Decoder = 5'd0;
373         to_IM = 1'd0;
374         to_DM = 1'd0;
375         to_PC = 1'd0;
376         to_AC = 3'b100;
377         en_RegSelector = 1'd1;
378         rw_RegSelector = 1'd0;

```

```

379         NS = INC6;
380     end
381     INC6: begin
382         to_ALU = 4'd0;
383         to_BUS = 5'd20;
384         to_Decoder = 5'd0;
385         to_IM = 1'd0;
386         to_DM = 1'd0;
387         to_PC = 1'd0;
388         to_AC = 3'b000;
389         en_RegSelector = 1'd1;
390         rw_RegSelector = 1'd1;
391         NS = FETCH1;
392     end
393     INCAC1: begin
394         to_ALU = 4'd7;
395         to_BUS = 5'd0;
396         to_Decoder = 5'd0;
397         to_IM = 1'd0;
398         to_DM = 1'd0;
399         to_PC = 1'd0;
400         to_AC = 3'b100;
401         en_RegSelector = 1'd0;
402         rw_RegSelector = 1'd0;
403         NS = FETCH1;
404     end
405     MOVE1: begin
406         to_ALU = 4'd0;
407         to_BUS = 5'd0;
408         to_Decoder = 5'd6;
409         to_IM = 1'd0;
410         to_DM = 1'd0;
411         to_PC = 1'd1;
412         to_AC = 3'b000;
413         en_RegSelector = 1'd0;
414         rw_RegSelector = 1'd0;
415         NS = MOVE2;
416     end
417     MOVE2: begin
418         to_ALU = 4'd1;
419         to_BUS = 5'd0;
420         to_Decoder = 5'd0;
421         to_IM = 1'd0;
422         to_DM = 1'd0;
423         to_PC = 1'd0;
424         to_AC = 3'b100;
425         en_RegSelector = 1'd1;
426         rw_RegSelector = 1'd0;
427         NS = MOVE3;
428     end
429     MOVE3: begin
430         to_ALU = 4'd0;
431         to_BUS = 5'd2;
432         to_Decoder = 5'd1;

```



```

433         to_IM = 1'd0;
434         to_DM = 1'd0;
435         to_PC = 1'd0;
436         to_AC = 3'b000;
437         en_RegSelector = 1'd0;
438         rw_RegSelector = 1'd0;
439         NS = MOVE4;
440     end
441     MOVE4: begin
442         to_ALU = 4'd0;
443         to_BUS = 5'd0;
444         to_Decoder = 5'd6;
445         to_IM = 1'd0;
446         to_DM = 1'd0;
447         to_PC = 1'd1;
448         to_AC = 3'b000;
449         en_RegSelector = 1'd0;
450         rw_RegSelector = 1'd0;
451         NS = MOVE5;
452     end
453     MOVE5: begin
454         to_ALU = 4'd0;
455         to_BUS = 5'd20;
456         to_Decoder = 5'd0;
457         to_IM = 1'd0;
458         to_DM = 1'd0;
459         to_PC = 1'd0;
460         to_AC = 3'b000;
461         en_RegSelector = 1'd1;
462         rw_RegSelector = 1'd1;
463         NS = FETCH1;
464     end
465     ADD: begin
466         to_ALU = 4'd2;
467         to_BUS = 5'd17;
468         to_Decoder = 5'd0;
469         to_IM = 1'd0;
470         to_DM = 1'd0;
471         to_PC = 1'd0;
472         to_AC = 3'b100;
473         en_RegSelector = 1'd0;
474         rw_RegSelector = 1'd0;
475         NS = FETCH1;
476     end
477     SUB: begin
478         to_ALU = 4'd3;
479         to_BUS = 5'd17;
480         to_Decoder = 5'd0;
481         to_IM = 1'd0;
482         to_DM = 1'd0;
483         to_PC = 1'd0;
484         to_AC = 3'b100;
485         en_RegSelector = 1'd0;
486         rw_RegSelector = 1'd0;

```

```

487         NS = FETCH1;
488     end
489     MUL: begin
490         to_ALU = 4'd4;
491         to_BUS = 5'd17;
492         to_Decoder = 5'd0;
493         to_IM = 1'd0;
494         to_DM = 1'd0;
495         to_PC = 1'd0;
496         to_AC = 3'b100;
497         en_RegSelector = 1'd0;
498         rw_RegSelector = 1'd0;
499         NS = FETCH1;
500     end
501     AND: begin
502         to_ALU = 4'd5;
503         to_BUS = 5'd17;
504         to_Decoder = 5'd0;
505         to_IM = 1'd0;
506         to_DM = 1'd0;
507         to_PC = 1'd0;
508         to_AC = 3'b100;
509         en_RegSelector = 1'd0;
510         rw_RegSelector = 1'd0;
511         NS = FETCH1;
512     end
513     OR: begin
514         to_ALU = 4'd6;
515         to_BUS = 5'd17;
516         to_Decoder = 5'd0;
517         to_IM = 1'd0;
518         to_DM = 1'd0;
519         to_PC = 1'd0;
520         to_AC = 3'b100;
521         en_RegSelector = 1'd0;
522         rw_RegSelector = 1'd0;
523         NS = FETCH1;
524     end
525     JUMP1: begin
526         to_ALU = 4'd0;
527         to_BUS = 5'd0;
528         to_Decoder = 5'd6;
529         to_IM = 1'd0;
530         to_DM = 1'd0;
531         to_PC = 1'd1;
532         to_AC = 3'b000;
533         en_RegSelector = 1'd0;
534         rw_RegSelector = 1'd0;
535         NS = JUMP2;
536     end
537     JUMP2: begin
538         to_ALU = 4'd0;
539         to_BUS = 5'd0;
540         to_Decoder = 5'd0;

```

```

541         to_IM = 1'd0;
542         to_DM = 1'd0;
543         to_PC = 1'd0;
544         to_AC = 3'b000;
545         en_RegSelector = 1'd0;
546         rw_RegSelector = 1'd0;
547         if((MDDR==16'd0 && z==1'b1) || (MDDR==16'd2 &&
i==1'b1) || (MDDR==16'd4 && j==1'b1) || (MDDR==16'd6 && k==1'b1) ||
(MDDR==16'd1 && z==1'b0) || (MDDR==16'd3 && i==1'b0) || (MDDR==16'd5 &&
j==1'b0) || (MDDR==16'd7 && k==1'b0)))
548             NS = JUMPY1;
549         else
550             NS = JUMPN1;
551     end
552     JUMPN1: begin
553         to_ALU = 4'd0;
554         to_BUS = 5'd0;
555         to_Decoder = 5'd0;
556         to_IM = 1'd0;
557         to_DM = 1'd0;
558         to_PC = 1'd1;
559         to_AC = 3'b000;
560         en_RegSelector = 1'd0;
561         rw_RegSelector = 1'd0;
562         NS = FETCH1;
563     end
564     JUMPY1: begin
565         to_ALU = 4'd0;
566         to_BUS = 5'd2;
567         to_Decoder = 5'd1;
568         to_IM = 1'd0;
569         to_DM = 1'd0;
570         to_PC = 1'd0;
571         to_AC = 3'b000;
572         en_RegSelector = 1'd0;
573         rw_RegSelector = 1'd0;
574         NS = JUMPY2;
575     end
576     JUMPY2: begin
577         to_ALU = 4'd0;
578         to_BUS = 5'd0;
579         to_Decoder = 5'd6;
580         to_IM = 1'd0;
581         to_DM = 1'd0;
582         to_PC = 1'd1;
583         to_AC = 3'b000;
584         en_RegSelector = 1'd0;
585         rw_RegSelector = 1'd0;
586         NS = JUMPY3;
587     end
588     JUMPY3: begin
589         to_ALU = 4'd0;
590         to_BUS = 5'd4;
591         to_Decoder = 5'd2;
592         to_IM = 1'd0;

```

```

593         to_DM = 1'd0;
594         to_PC = 1'd0;
595         to_AC = 3'b000;
596         en_RegSelector = 1'd0;
597         rw_RegSelector = 1'd0;
598         NS = FETCH1;
599     end
600     STORM1: begin
601         to_ALU = 4'd0;
602         to_BUS = 5'd7;
603         to_Decoder = 5'd1;
604         to_IM = 1'd0;
605         to_DM = 1'd0;
606         to_PC = 1'd0;
607         to_AC = 3'b000;
608         en_RegSelector = 1'd0;
609         rw_RegSelector = 1'd0;
610         NS = STORM2;
611     end
612     STORM2: begin
613         to_ALU = 4'd0;
614         to_BUS = 5'd20;
615         to_Decoder = 5'd4;
616         to_IM = 1'd0;
617         to_DM = 1'd1;
618         to_PC = 1'd0;
619         to_AC = 3'b000;
620         en_RegSelector = 1'd0;
621         rw_RegSelector = 1'd0;
622         NS = FETCH1;
623     end
624 endcase
625 end
626 endmodule

```