

```

1 Control signals for various states:
2 // This is our default state. It assigns instruction address to memory
3 State 8'h0:
4         ld_mar=0;
5         ld_pc=0;
6         reg_r=0;
7         reg_w=0;
8         pc_alu=1;
9         ld_mar=1;
10        mem_size=1;
11        mem_r_w=1;
12
13 // Read state. Reads instruction from memory to mdr
14 State 8'h1:
15        ld_mdr=1;
16        mdr_sext=0;
17        ld_mar=0;
18        ld_ir=0;
19        ld_pc=1;
20        reg_r=0;
21        reg_w=0;
22        mem_r_w=1;
23        reset=0;
24
25 // Fetch state. IR fetches instruction
26 State 8'h2:
27        ld_mar=1;
28        ld_pc=0;
29        reg_r=0;
30        reg_w=0;
31        mem_r_w=1;
32        mar_8=0;
33        sel_mem=01;
34        mem_r_w=1;
35        ld_ir=1;
36
37 // Load state. Loads in ir
38 State 8'h3:
39        ld_mar=0;
40        ld_ir=1;
41        ld_pc=0;
42        reg_r=0;
43        reg_w=0;
44        mem_r_w=1;
45
46 // Decode state. This decodes the instruction
47 State 8'h4:
48        ld_mar=0;
49        ld_ir=0;
50        ld_pc=0;
51        reg_r=0;
52        reg_w=0;
53        mem_r_w=1;
54
55 // Once we have decoded the instruction, we can start with the signals
56
57 // BR instruction state i.e Opcode [15:12] = 0000
58 State 8'h10:
59        ld_mdr=0;
60        ld_mar=0;

```

```

61         ld_ir=0;
62         ld_pc=0;
63         reg_r=0;
64         reg_w=0;
65         mem_r_w=1;
66         if((n==opcode[11]) or (z==opcode[10]) or (p==opcode[9]))
67         {
68             nine=01;
69             sel_pc=01;
70         }
71 State 8'h11:
72         ld_mdr=0;
73         ld_mar=0;
74         ld_ir=0;
75         reg_r=0;
76         reg_w=0;
77         mem_r_w=1;
78         ld_pc=0;
79
80 // ADD instruction state i.e Opcode [15:12] = 0001
81 State 8'h20:
82         ld_mdr=0;
83         ld_mar=0;
84         ld_ir=0;
85         ld_pc=0;
86         mem_r_w=1 ;
87         sr1 = opcode[8:6];
88         sr2 = opcode[2:0];
89         dr = opcode[11:9];
90         reg_r=1;
91         reg_w=0;
92         alusel={opcode[5],1'b0};
93
94 State 8'h21:
95         ld_mdr=0;
96         ld_mar=0;
97         ld_ir=0;
98         ld_pc=0;
99         reg_r=0;
100        reg_w=0;
101        mem_r_w=1;
102        opsel=001;
103        reg_w=1;
104        reg_r=0;
105        rfsel=11;
106        sel_pc=2'b10;
107
108 // LDB instruction state i.e Opcode [15:12] = 0010
109 State 8'h30:
110        pc_alu=0;
111        ld_mdr=0;
112        ld_mar=0;
113        ld_ir=0;
114        ld_pc=0;
115        mem_r_w=1;
116        reg_r=1;
117        reg_w=0;
118        alusel=01;
119
120        dr=opcode[11:9];
121        sr1=opcode[8:6];

```

```

122
123 State 8'h31:
124     ld_mdr=0;
125     ld_mar=0;
126     ld_ir=0;
127     ld_pc=0;
128     reg_r=0;
129     reg_w=0;
130     mem_r_w=1;
131     opsel=001;
132     ld_mar=1;
133     mem_size=0;
134
135 State 8'h32:
136     ld_mar=0;
137     ld_ir=0;
138     ld_pc=0;
139     mem_r_w=1;
140     ld_mdr=1;
141     mdr_sext=1;
142     sel_mem=01;
143     rfsel=00;
144     reg_w=1;
145     reg_r=0;
146     sel_pc=2'b10;
147
148 // STB instruction state i.e Opcode [15:12] = 0011
149 State 8'h40:
150     pc_alu=0;
151     ld_mdr=0;
152     ld_mar=0;
153     ld_ir=0;
154     ld_pc=0;
155     mem_r_w=1;
156     reg_r=1;
157     reg_w=0;
158     alusel=01;
159     opsel=001;
160
161     sr2=opcode[11:9];
162     sr1=opcode[8:6];
163
164 State 8'h41:
165     ld_mdr=0;
166     ld_ir=0;
167     ld_pc=0;
168     reg_r=0;
169     reg_w=0;
170     mem_r_w=1;
171     pc_alu=0;
172     ld_mar=1;
173     mem_size=0;
174
175 State 8'h42:
176     ld_mdr=1;
177     ld_ir=0;
178     ld_pc=0;
179     reg_r=0;
180     reg_w=0;
181     mem_r_w=0;
182     pc_alu=0;

```

```

183         mdr_sext=0;
184         ld_mar=0;
185         sel_pc=2'b10;
186         sel_mem=10;
187
188 // JSR/JSRR instruction state i.e Opcode [15:12] = 0100
189 State 8'h50:
190         ld_mdr=0;
191         ld_mar=0;
192         ld_ir=0;
193         ld_pc=0;
194         mem_r_w=1;
195         dr=3'd7;
196         reg_w=1;
197         reg_r=0;
198         rfsel=10;
199
200         sr1=opcode[8:6];
201
202 State 8'h51:
203         ld_mdr=0;
204         ld_mar=0;
205         ld_ir=0;
206         ld_pc=0;
207         reg_w=0;
208         mem_r_w=1;
209         if(opcode[11]==1)
210         {   reg_r=0;
211             nine=00;
212             sel_pc=01;
213         }
214         else
215         {   reg_r=1;
216             alusel=01;
217             opsel=001;
218             sel_pc=00;
219         }
220
221 State 8'h52:
222         alusel=10;
223         sel_pc=00;
224         ld_mdr=0;
225         ld_mar=0;
226         ld_ir=0;
227         ld_pc=1;
228         reg_r=0;
229         reg_w=1;
230         mem_r_w=1;
231
232 State 8'h53:
233         sel_pc=10;
234         ld_mdr=0;
235         ld_mar=0;
236         ld_ir=0;
237         ld_pc=1;
238         reg_r=0;
239         reg_w=0;
240         mem_r_w=1;
241
242 // AND instruction state i.e Opcode [15:12] = 0101
243 State 8'h60:

```

```

244         ld_mdr=0;
245         ld_mar=0;
246         ld_ir=0;
247         ld_pc=0;
248         mem_r_w=1;
249         reg_r=1;
250         reg_w=0;
251         alusel={opcode[5],1'b0};
252
253         sr1 = opcode[8:6];
254         sr2 = opcode[2:0];
255         dr = opcode[11:9];
256
257     State 8'h61:
258         ld_mdr=0;
259         ld_mar=0;
260         ld_ir=0;
261         ld_pc=0;
262         mem_r_w=1;
263         opsel=000;
264         reg_w=1;
265         reg_r=0;
266         rfsel=11;
267         sel_pc=2'b10;
268
269     // LDW instruction state i.e Opcode [15:12] = 0110
270     State 8'h70:
271         pc_alu=0;
272         ld_mdr=0;
273         ld_mar=0;
274         ld_ir=0;
275         ld_pc=0;
276         mem_r_w=1;
277         reg_r=1;
278         reg_w=0;
279         alusel=01;
280
281         dr=opcode[11:9];
282         sr1=opcode[8:6];
283
284     State 8'h71:
285         ld_mdr=0;
286         ld_ir=0;
287         ld_pc=0;
288         reg_r=0;
289         reg_w=0;
290         opsel=001;
291         pc_alu=0;
292         ld_mar=1;
293         mem_size=1;
294         mem_r_w=1;
295
296     State 8'h72:
297         ld_mar=0;
298         ld_ir=0;
299         ld_pc=0;
300         mem_r_w=1;
301         ld_mdr=1;
302         mdr_sext=0;
303         sel_mem=01;
304         rfsel=00;

```

```

305         reg_w=1;
306         reg_r=0;
307         sel_pc=2'b10;
308
309 // STW instruction state i.e Opcode [15:12] = 0111
310 State 8'h80:
311         pc_alu=0;
312         ld_mdr=0;
313         ld_mar=0;
314         ld_ir=0;
315         ld_pc=0;
316         mem_r_w=1;
317         reg_r=1;
318         reg_w=0;
319         alusel=01;
320
321         sr2=opcode[11:9];
322         sr1=opcode[8:6];
323
324 State 8'h81:
325         ld_mdr=0;
326         ld_ir=0;
327         ld_pc=0;
328         reg_r=0;
329         reg_w=0;
330         pc_alu=0;
331         ld_mar=1;
332         mem_size=1;
333         sel_mem=10;
334         mem_r_w=1;
335
336 State 8'h82:
337         ld_mdr=1;
338         ld_ir=0;
339         ld_pc=0;
340         reg_r=0;
341         reg_w=0;
342         mem_r_w=0;
343         pc_alu=0;
344         mdr_sext=0;
345         ld_mar=0;
346         sel_mem=10;
347         sel_pc=2'b10;
348
349 //XOR instruction state i.e Opcode [15:12] = 1001
350 State 8'h90:
351         ld_mdr=0;
352         ld_mar=0;
353         ld_ir=0;
354         ld_pc=0;
355         mem_r_w=1;
356         reg_r=1;
357         reg_w=0;
358         alusel={opcode[5],1'b0};
359
360 State 8'h91:
361         ld_mdr=0;
362         ld_mar=0;
363         ld_ir=0;
364         ld_pc=0;
365         reg_r=0;

```

```

366         reg_w=0;
367         mem_r_w=1;
368         opsel=010;
369
370 State 8'h92:
371         ld_mdr=0;
372         ld_mar=0;
373         ld_ir=0;
374         ld_pc=0;
375         mem_r_w=1;
376         reg_w=1;
377         reg_r=0;
378         rfsel=11;
379         sel_pc=2'b10;
380
381 //JMP/RET instruction state i.e Opcode [15:12] = 1100
382 State 8'ha0:
383
384         ld_mdr=0;
385         ld_mar=0;
386         ld_ir=0;
387         ld_pc=0;
388         mem_r_w=1;
389         reg_r=1;
390         reg_w=0;
391         alusel=10;
392         opsel=001;
393         sel_pc=00;
394
395         sr1=opcode[8:6];
396
397 State 8'ha1:
398         ld_mdr=0;
399         ld_mar=0;
400         ld_ir=0;
401         ld_pc=1;
402         mem_r_w=1;
403         reg_r=1;
404         reg_w=0;
405
406 // LEA instruction state i.e Opcode [15:12] = 1110
407 State 8'hc0:
408         ld_mdr=0;
409         ld_mar=0;
410         ld_ir=0;
411         ld_pc=0;
412         reg_r=0;
413         reg_w=0;
414         mem_r_w=1;
415         nine=01;
416
417         dr=opcode[11:9];
418
419 State 8'hc1:
420         ld_mdr=0;
421         ld_mar=0;
422         ld_ir=0;
423         ld_pc=0;
424         mem_r_w=1;
425         reg_w=1;
426         reg_r=0;

```

```

427         rfsel=01;
428         sel_pc=2'b10;
429
430 //SHF instruction state i.e Opcode [15:12] = 1101
431 State 8'hb0:
432     ld_mdr=0;
433     ld_mar=0;
434     ld_ir=0;
435     ld_pc=0;
436     mem_r_w=1;
437     reg_r=1;
438     reg_w=0;
439     alusel=11;
440
441     sr1 = opcode[8:6];
442     dr = opcode[11:9];
443
444 State 8'hb1:
445     ld_mdr=0;
446     ld_mar=0;
447     ld_ir=0;
448     ld_pc=0;
449     mem_r_w=1;
450     reg_w=1;
451     reg_r=0;
452     rfsel=11;
453
454     if (opcode[5:4] == 2'b00) opsel=011;
455     else if (opcode[5:4] == 2'b01) opsel=100;
456     else if (opcode[5:4] == 2'b10) opsel=101;

```
