

```

1 #include "simulator.h"
2
3
4
5 void run(int * program_begin, int * program_end, void * memory, cpu_t * cpu)
6 {
7     printf("%X %X\n", *program_begin, *program_end);
8     cpu->PC = *program_begin;
9     int temp = 0;
10    int temp2 = 0;
11    int temp3 = 0;
12    int x_mod = 0;
13    FILE * input = NULL;
14    FILE * output = NULL;
15    input = fopen("F1.txt", "r");
16    output = fopen("F5.txt", "w");
17    cpu->L = *program_end;
18
19
20    while(cpu->PC < *program_end)
21    {
22        x_mod = 0;
23        temp = 0;
24        temp2 = 0;
25        temp3 = 0;
26        if(*(uint8_t *)memory + cpu->PC + 1) >= 128)
27        {
28            x_mod = 1;
29        }
30
31        switch(*(uint8_t *)memory + cpu->PC)
32        {
33            case 24:
34                printf("ADD      ");
35                temp = *(uint8_t *)memory + cpu->PC + 1);
36                temp = temp << 8;
37                temp |= *(uint8_t *)memory + cpu->PC + 2);
38
39                temp2 = *(uint8_t *)memory + temp + 2);
40                cpu->A += temp2;
41
42                temp2 = *(uint8_t *)memory + temp + 1);
43                temp2 = temp2 << 8;
44                cpu->A += temp2;
45
46                temp2 = *(uint8_t *)memory + temp);
47                temp2 = temp2 << 16;
48                cpu->A += temp2;
49
50
51                cpu->PC += 3;
52                break;
53
54            case 52:
55                printf("JGT      ");
56                cpu->PC += 3;
57                break;
58
59            case 0:
60                printf("LDA      ");
61                temp = *(uint8_t *)memory + cpu->PC + 1);
62                temp = temp << 8;
63                temp |= *(uint8_t *)memory + cpu->PC + 2);
64
65                if(*(uint8_t *)memory + temp) == '.')
66                {
67                    cpu->A = 0;
68                }
69                else
70                {
71                    cpu->A = *(uint8_t *)memory + temp);
72                }
73
74                cpu->A = cpu->A << 8;
75
76                if(*(uint8_t *)memory + temp + 1) == '.')
77                {
78                    cpu->A |= 0;

```

```

79     }
80     else
81     {
82         cpu->A |= *((uint8_t *)memory + temp + 1);
83     }
84
85     cpu->A = cpu->A << 8;
86
87     if(*((uint8_t *)memory + temp + 2) == '.')
88     {
89         cpu->A |= 0;
90     }
91     else
92     {
93         cpu->A |= *((uint8_t *)memory + temp + 2);
94     }
95
96
97
98     cpu->PC += 3;
99     break;
100
101 case 4:
102     printf("LDX      ");
103     temp = *((uint8_t *)memory + cpu->PC + 1);
104     temp = temp << 8;
105     temp |= *((uint8_t *)memory + cpu->PC + 2);
106
107     if(*((uint8_t *)memory + temp) == '.')
108     {
109         cpu->X = 0;
110     }
111     else
112     {
113         cpu->X = *((uint8_t *)memory + temp);
114     }
115
116     cpu->X = cpu->X << 8;
117
118     if(*((uint8_t *)memory + temp + 1) == '.')
119     {
120         cpu->X |= 0;
121     }
122     else
123     {
124         cpu->X |= *((uint8_t *)memory + temp + 1);
125     }
126
127     cpu->X = cpu->X << 8;
128
129     if(*((uint8_t *)memory + temp + 2) == '.')
130     {
131         cpu->X |= 0;
132     }
133     else
134     {
135         cpu->X |= *((uint8_t *)memory + temp + 2);
136     }
137
138
139     cpu->PC += 3;
140     break;
141
142 case 216:
143     printf("RD      ");
144     temp = *((uint8_t *)memory + cpu->PC + 1);
145     temp = temp << 8;
146     temp |= *((uint8_t *)memory + cpu->PC + 2);
147
148
149     //cpu->A = *((uint8_t *)memory + temp);
150
151     temp2 = getc(input);
152     if(temp2 == -1)
153     {
154         cpu->A = 0;
155     }
156     else
157     {

```

```

158         cpu->A = (unsigned int)temp2;
159     }
160
161
162
163     cpu->PC += 3;
164     break;
165
166 case 220:
167     printf("WD      ");
168     temp = *((uint8_t *)memory + cpu->PC + 1);
169     temp = temp << 8;
170     temp |= *((uint8_t *)memory + cpu->PC + 2);
171
172     /*((uint8_t *)memory + temp) = cpu->A;
173     putc(cpu->A, output);
174
175     cpu->PC += 3;
176     break;
177
178 case 84:
179     printf("STCH    ");
180     temp = *((uint8_t *)memory + cpu->PC + 1);
181     temp = temp << 8;
182     temp |= *((uint8_t *)memory + cpu->PC + 2);
183
184     if(x_mod == 1)
185     {
186         temp = temp ^ (1 << 15);
187         *((uint8_t *)memory + temp + cpu->X) = cpu->A;
188     }
189     else
190     {
191         *((uint8_t *)memory + temp) = cpu->A;
192     }
193
194     cpu->PC += 3;
195     break;
196
197 case 16:
198     printf("STX      ");
199     temp = *((uint8_t *)memory + cpu->PC + 1);
200     temp = temp << 8;
201     temp |= *((uint8_t *)memory + cpu->PC + 2);
202
203     temp2 = cpu->X;
204
205     *((uint8_t *)memory + temp + 2) = temp2;
206     temp2 = temp2 >> 8;
207     *((uint8_t *)memory + temp + 1) = temp2;
208     temp2 = temp2 >> 8;
209     *((uint8_t *)memory + temp) = temp2;
210
211
212     cpu->PC += 3;
213
214     break;
215
216 case 12:
217     printf("STA      ");
218     temp = *((uint8_t *)memory + cpu->PC + 1);
219     temp = temp << 8;
220     temp |= *((uint8_t *)memory + cpu->PC + 2);
221
222     temp2 = cpu->A;
223
224     *((uint8_t *)memory + temp + 2) = temp2;
225     temp2 = temp2 >> 8;
226     *((uint8_t *)memory + temp + 1) = temp2;
227     temp2 = temp2 >> 8;
228     *((uint8_t *)memory + temp) = temp2;
229
230
231     cpu->PC += 3;
232
233     break;
234
235 case 44:
236     printf("TIX      ");

```

```

237     cpu->X++;
238
239     temp = *((uint8_t *)memory + cpu->PC + 1);
240     temp = temp << 8;
241     temp |= *((uint8_t *)memory + cpu->PC + 2);
242
243     temp2 = *((uint8_t *)memory + temp);
244     temp2 = temp2 << 8;
245     temp2 |= *((uint8_t *)memory + temp + 1);
246     temp2 = temp2 << 8;
247     temp2 |= *((uint8_t *)memory + temp + 2);
248
249     if(cpu->X == temp2)
250     {
251         cpu->SW = '=';
252     }
253     else if(cpu->X < temp2)
254     {
255         cpu->SW = '<';
256     }
257     else
258     {
259         cpu->SW = '>';
260     }
261
262     cpu->PC += 3;
263     break;
264
265 case 64:
266     printf("AND      ");
267     cpu->PC += 3;
268     break;
269
270 case 60:
271     printf("J      ");
272     temp = cpu->PC;
273     cpu->PC = *((uint8_t *)memory + temp + 1);
274     cpu->PC = cpu->PC << 8;
275     cpu->PC |= *((uint8_t *)memory + temp + 2);
276     break;
277
278 case 56:
279     printf("JLT      ");
280     temp = cpu->PC;
281     if(cpu->SW == '<')
282     {
283         cpu->PC = *((uint8_t *)memory + temp + 1);
284         cpu->PC = cpu->PC << 8;
285         cpu->PC |= *((uint8_t *)memory + temp + 2);
286     }
287     else
288     {
289         cpu->PC += 3;
290     }
291     break;
292
293 case 80:
294     printf("LDCH      ");
295     temp = *((uint8_t *)memory + cpu->PC + 1);
296     temp = temp << 8;
297     temp |= *((uint8_t *)memory + cpu->PC + 2);
298
299     if(x_mod == 1)
300     {
301         temp = temp ^ (1 << 15);
302         cpu->A = *((uint8_t *)memory + temp + cpu->X);
303     }
304     else
305     {
306         cpu->A = *((uint8_t *)memory + temp);
307     }
308
309     cpu->PC += 3;
310     break;
311
312 case 32:
313     printf("MUL      ");
314

```

```

316     temp = *((uint8_t *)memory + cpu->PC + 1);
317     temp = temp << 8;
318     temp |= *((uint8_t *)memory + cpu->PC + 2);
319
320     temp3 = cpu->A;
321     cpu->A = 0;
322
323     temp2 = *((uint8_t *)memory + temp + 2);
324     cpu->A += temp3 * temp2;
325
326     temp2 = *((uint8_t *)memory + temp + 1);
327     temp2 = temp2 << 8;
328     cpu->A += temp3 * temp2;
329
330     temp2 = *((uint8_t *)memory + temp);
331     temp2 = temp2 << 16;
332     cpu->A += temp3 * temp2;
333
334
335     cpu->PC += 3;
336     break;
337
338 case 76:
339     printf("RSUB    ");
340     cpu->PC = cpu->L;
341     break;
342
343 case 20:
344     printf("STL      ");
345     temp = *((uint8_t *)memory + cpu->PC + 1);
346     temp = temp << 8;
347     temp |= *((uint8_t *)memory + cpu->PC + 2);
348
349     temp2 = cpu->L;
350
351     *((uint8_t *)memory + temp + 2) = temp2;
352     temp2 = temp2 >> 8;
353     *((uint8_t *)memory + temp + 1) = temp2;
354     temp2 = temp2 >> 8;
355     *((uint8_t *)memory + temp) = temp2;
356
357
358
359     cpu->PC += 3;
360     break;
361
362 case 28:
363     printf("SUB      ");
364     temp = *((uint8_t *)memory + cpu->PC + 1);
365     temp = temp << 8;
366     temp |= *((uint8_t *)memory + cpu->PC + 2);
367
368     temp2 = *((uint8_t *)memory + temp);
369     temp2 = temp2 << 8;
370
371     temp2 = *((uint8_t *)memory + temp + 1);
372     temp2 = temp2 << 8;
373
374     temp2 = *((uint8_t *)memory + temp + 2);
375
376     cpu->A = cpu->A - temp2;
377
378
379     cpu->PC += 3;
380     break;
381
382
383
384 case 40:
385     printf("COMP     ");
386     temp = *((uint8_t *)memory + cpu->PC + 1);
387     temp = temp << 8;
388     temp |= *((uint8_t *)memory + cpu->PC + 2);
389
390     temp2 = *((uint8_t *)memory + temp);
391     temp2 = temp2 << 8;
392     temp2 |= *((uint8_t *)memory + temp + 1);
393     temp2 = temp2 << 8;
394     temp2 |= *((uint8_t *)memory + temp + 2);

```

```

395
396     if(cpu->A == temp2)
397     {
398         cpu->SW = '=';
399     }
400     else if(cpu->A < temp2)
401     {
402         cpu->SW = '<';
403     }
404     else
405     {
406         cpu->SW = '>';
407     }
408
409     cpu->PC += 3;
410     break;
411
412 case 48:
413     printf("JEQ      ");
414     temp = cpu->PC;
415     if(cpu->SW == '=')
416     {
417         cpu->PC = *((uint8_t *)memory + temp + 1);
418         cpu->PC = cpu->PC << 8;
419         cpu->PC |= *((uint8_t *)memory + temp + 2);
420
421     }
422     else
423     {
424         cpu->PC += 3;
425     }
426     break;
427
428 case 72:
429     printf("JSUB      ");
430     temp = cpu->PC;
431     cpu->L = cpu->PC + 3;
432     cpu->PC = *((uint8_t *)memory + temp + 1);
433     cpu->PC = cpu->PC << 8;
434     cpu->PC |= *((uint8_t *)memory + temp + 2);
435     break;
436
437 case 8:
438     printf("LDL      ");
439     temp = *((uint8_t *)memory + cpu->PC + 1);
440     temp = temp << 8;
441     temp |= *((uint8_t *)memory + cpu->PC + 2);
442
443     if(*((uint8_t *)memory + temp) == '.')
444     {
445         cpu->L = 0;
446     }
447     else
448     {
449         cpu->L = *((uint8_t *)memory + temp);
450     }
451
452     cpu->L = cpu->L << 8;
453
454     if(*((uint8_t *)memory + temp + 1) == '.')
455     {
456         cpu->L |= 0;
457     }
458     else
459     {
460         cpu->L |= *((uint8_t *)memory + temp + 1);
461     }
462
463     cpu->L = cpu->L << 8;
464
465     if(*((uint8_t *)memory + temp + 2) == '.')
466     {
467         cpu->L |= 0;
468     }
469     else
470     {
471         cpu->L |= *((uint8_t *)memory + temp + 2);
472     }
473

```

```

474
475
476     cpu->PC += 3;
477     break;
478
479 case 68:
480     printf("OR      ");
481     cpu->PC += 3;
482     break;
483
484 case 232:
485     printf("STSW    ");
486     cpu->PC += 3;
487     break;
488
489 case 224:
490     printf("TD      ");
491     temp = *((uint8_t *)memory + cpu->PC + 1);
492     temp = temp << 8;
493     temp |= *((uint8_t *)memory + cpu->PC + 2);
494
495
496     temp2 = *((uint8_t *)memory + temp);
497     //printf("%02X\n", temp2);
498
499     if(temp2 == hex_to_dex("F1"))
500     {
501         if(input != NULL)
502         {
503             cpu->SW = '<';
504         }
505         else
506         {
507             cpu->SW = '=';
508         }
509     }
510
511     if(temp2 == hex_to_dex("05"))
512     {
513         if(output != NULL)
514         {
515             cpu->SW = '<';
516         }
517         else
518         {
519             cpu->SW = '=';
520         }
521     }
522
523     /*
524     if(*((uint8_t *)memory + temp) == '.')
525     {
526         cpu->SW = '=';
527     }
528     else
529     {
530         cpu->SW = '<';
531     }
532     */
533
534     cpu->PC += 3;
535     break;
536
537 default:
538     break;
539 }
540 printf("A : %06X  X : %06X  L : %06X  PC : %06X  SW : %06X\n\n", cpu->A, cpu->X,
cpu->L, cpu->PC, cpu->SW);
541 }
542 fclose(input);
543 fclose(output);
544
545 printf("\n");
546 }
547

```