Simple Processor

Ethan Katzenberg - ek1295

Design decisions

- Program memory and data memory are adjacent, accessed via different busses

- (So far) No stack pointer, likely will be added in the next iteration

ISA

- Arithmetic - Add, Sub, Mul, Div
- Registers - Mov
- Memory - Ldr, Str
- Control Flow - Jmp, Brc
- Conditionals - Blt(i), Beq(i), Cmp

Interface

Vector Sort GCD Step Run

```
start
        mov $r0, #0
        ldr $r1, numItems
        mov $r2, #0
        mov $r10, #1
loopStart
        mov $r0, #0
        sub $r1, $r1, #1
        beq end, $r1, $r2
innerLoop
        beq loopStart, $r0, $r1
        add $r3, $r0, $r10
        ldr $r4, [array, $r0]
        ldr $r5, [array, $r3]
        brc swap
        add $r0, $r0, #1
        jmp innerLoop
end
        hlt
swap
        blt $r30, $r4, $r5
        str $r4, [array, $r3]
        str $r5, [array, $r0]
        jmp $r30
numItems
        .word 14
array
        .word 15,22, 3, 19, 55, 11, 23, -4, 5, 6, 14, 15, 23,
55
```

Memory	Current Instruction
76:14	hlt
80:-4	r0 0
84:3	r1 0
88:5	r2 0
92:6	r3 1
96:11	r4 -4
100:14	r5 3
104:15	r6 0
108:15	r7 0
112:19	r8 0
116:22	r9 0
120:23	r10 1
124:23	r11 0
128:55	r12 0
132:55	r13 0
	r14 0
	r15 0
	r16 0
	r17 0
	r18 0
	r19 0

Execution instructions

- Open the index.htm file (preferably in Safari as programs are loaded in via ajax, and Chrome has an issue about ajax requests to local files)
- Select one of the attached programs
- Press step to step through the program, run to execute until hlt

Implemented algorithms (so far)

- Vector addition (asm/vector.asm)
- GCD (asm/gdc.asm)
- Bubble sort (asm/sort.asm)

Questionable Design Decisions

- Parameters within most instructions can take the form of either a register or immediate value (eg. MOV $r0, #10 or MOV $r0, $r1)

- Offsets are specified in increments of instruction length (e.g LDR $r0, [label, #0] and LDR $r0, [label, #1] will access memory 4 bytes apart on a 32bit machine)