P-Machine Simulator Display:

Registers				
pc	**mp**	**sp**	**ep**	**np**
16	**14**	**20**	**30**	

Instruction Store				Data Store				
Address	**Opcode**	**Operand 1**	**Operand 2**	**Sub Program**	**Address**	**Id**	**Type**	**Value**
0	ent	sp	7		**35**			
1	ent	ep	16		**34**			
2	lvi	0	6		**33**			
3	ldc	i	0		**32**			
4	equ	i			**31**			
5	fjp		10		**30**			
6	lda	0	0		**29**			
7	lvi	0	5		**28**			
8	sti	i			**27**			
9	ujp		18		**26**			
10	lda	0	0		**25**			
11	mst	1			**24**			
12	lvi	0	6		**23**			
13	lvi	0	5		**22**			
14	lvi	0	6		**21**			
15	mod				**20**	**b**	**int**	**24**
16	cup	2	0		**19**	**a**	**int**	**36**
17	sti	i			**18**	**ra**	**int**	**17**
18	rtn	i		**gcd**	**17**	**ep**	**int**	**30**
19	ent	sp	7		**16**	**dl**	**int**	**7**
20	ent	ep	14		**15**	**sl**	**int**	**0**
21	lda	0	5		**14**	**rv**	**int**	
22	ldc	96			**13**	**b**	**int**	**36**
23	sti	i			**12**	**a**	**int**	**96**
24	lda	0	6		**11**	**ra**	**int**	**17**
25	ldc	36		**gcd**	**10**	**ep**	**int**	**23**
26	sti	i			**9**	**dl**	**int**	**0**
27	mst	0			**8**	**sl**	**int**	**0**
28	lvi	0	5		**7**	**rv**	**int**	
29	lvi	0	6		**6**	**y**	**int**	**36**
30	cup	2	0		**5**	**x**	**int**	**96**
31	csp		wri		**4**	**ra**	**int**	**35**
32	rtn			**example**	**3**	**ep**	**int**	**14**
33	mst	0			**2**	**dl**	**int**	**0**
34	cup	0	19		**1**	**sl**	**int**	**0**
35	stp				**0**	**rv**		

Figure 1. P-Machine Simulator Display

Figure 1 is a snapshot of the display produced by the P-Machine Simulator for the Pascal program example shown in figure 2.

```
program example(input,output);
    var x,y: integer;
    function gcd(a,b:integer):integer;
    begin{gcd}
        if b=0 then gcd:=a
        else gcd:=gcd(b,a mod b)
    end{gcd};
begin{example}
    x:=96; y:=36;
    writeinteger(gcd(x,y))
end{example}.
```

Figure 2. Pascal program example

The P-Machine Visual Simulator employs a P-Machine to execute a P-Code program while at the same time displaying the symbolic form of the P-Code program and activation records created by the program.

The reader can find a specification of a P-Machine at http://cs2.uco.edu/~trt/cs4173/pspec.pdf. This document discusses how an executing P-Code program is to be displayed. The display has three parts, a register display, an instruction store, and a data store.

Register Display: The register display is shown at the top of the screen and contains five values, one for each of the registers. The registers are the program counter (**pc**), the mark pointer (**mp**), the stack pointer (**sp**), the extreme pointer (**ep**), and the new pointer (**np**).

Instruction Store Display: The Instruction Store display consists of four columns labeled **Address**, **Opcode**, **Operand 1**, and **Operand 2**. The Address column enumerates the sequential addresses of the P-Code program. Instructions in the instruction store are displayed such that the next instruction to be executed, the instruction whose address is stored in the program counter (pc) register, is in the middle of the display. The Opcode column contains the P-Code mnemonics of the instruction whose address is given in the first column of the row. Likewise, the columns labeled Operand 1 and Operand 2 contain the values of the first and second operands of the instruction in the row. Where possible, values for the operands are given as symbols or mnemonics. For example, operand 1 of an entry (ent) instruction is either the stack pointer register (sp) or the extreme pointer register (ep) and for that instruction operand 1 is displayed either as sp or ep. The reader may inspect the entry instructions located at addresses 0, 1, 19, and 20. In a similar way, various instructions refer to P-Machine types: these, too, are displayed symbolically as a – address, b – Boolean, c – character, I – integer, r – real, s – string, t –

set, p – procedure, and x – any. Instructions located at addresses 3 and 4 are examples of instructions that employ a type in operand 1.

Data Store Display: The Data Store display consists of five columns. The leftmost column labeled **Subprogram** delineates the activation records of the program. Each function, procedure, or the main program creates an activation record and the name of the subprogram that created the activation record is placed in the left column marking the rows of the activation. The next column is labeled **Address**. Addresses of the data store are numbered sequentially starting at zero shown at the bottom of the display and incrementing sequentially toward the top of the display. The column labeled **Id** contains the names of Pascal program variables or the P-Machine-controlled variables shown in the stack mark portion of the activation record. Not shown in the example but nonetheless displayed are compiler-generated temporaries and the computation stack. In the column labeled **Type** are types of the values whose names are shown in the **Id** column. The column labeled **Value** contains the values of the corresponding identifiers given in the **Id** column.