

Project: Create and print individual instructions in the trace file (.atrc). The instruction trace from file p00.atrc is shown in Figure 3 as an example. Please note that the instructions shown in Figure 3 are excerpted from file p00.atrc that contains a trace produced by the scanner and parser in addition to the instruction trace. The entire contents of file p00.atrc are shown in Figure 4.

Omit instructions that reference a label including **cup**, **ent**, **ujp**, **fjp**, **tjp**, and **xjp**. Omit processing the LABEL and DEFINE tokens. Omit the **ldc** instruction.

Program Files:	File	Description
	makepasm	File makepasm contains instructions for program pasm . Instructions are written for the <i>Unix</i> utility make . Program pasm is contained in file pasm .
	mkpasm	File mkpasm is a Unix script file that removes old file created in the last creation of executable file pas and invokes file makepasm to create a new executable file pasm .

Command Line: Project **p04.3** can be invoked with zero or one program parameters. The first program parameter is the input file name. Sample command lines together with corresponding actions by program **pasm** are shown below. Boldfaced type indicates data entered at the keyboard by the user.

\$ pasm
Enter the input file name: **p00.pasm**

\$ pasm p00.pasm

Input File: The input file contains a P-Code Assembler program. The input file name must have the suffix ".pcd" Please refer to figure 1 for an example of the format of an input file.

Output Files: A single output file is produced. The output file has the same prefix as the input **source.pcd** and the suffix **.pcd** is replaced by the suffix **.atrc**. For example, if the input file was named **hello.pcd**, the output file would be named **hello.atrc**.

\$ pasm p00.pcd

File **p00.atrc** is produced as shown in Figure 2.

L00001	ent	sp	L00002
	ent	ep	L00003
	rtn	p	
#define	L00002	4	
#define	L00003	4	
	mst	0	
	cup	0	L00001
	stp		

Figure 1. Example input file p00.pcd.

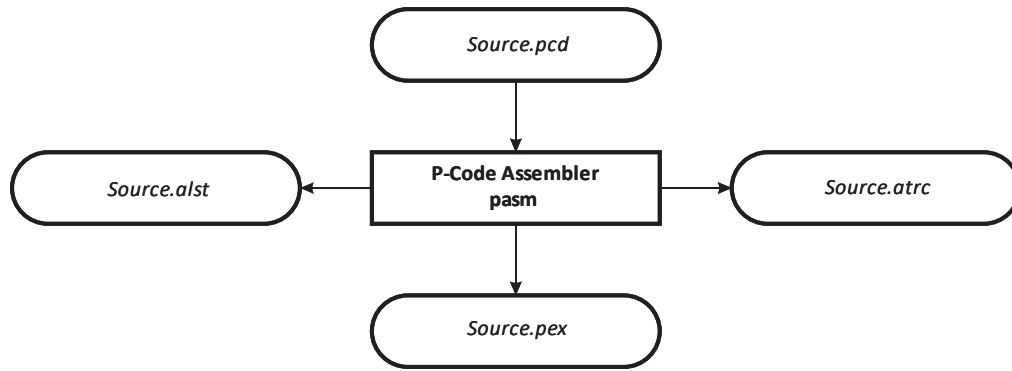


Figure 2. P-Code Assembler Block Diagram

rtn(04)	p(07)	0(0000)
mst(03)	0(00)	0(0000)
stp(30)	0(00)	0(0000)

Figure 3. Excerpts from File p00.atrc

```
Token=(258,LABEL,L00001)
#009 label-list->LABEL
Token=(268,ENT_0,ent)
Token=(338,SP_R,sp)
#080 register->SP
Token=(258,LABEL,L00002)
060 class2-operation->ent register INTLIT
014 operation->class3-operation
#007 statement->label-list operation
#004 item->statement
#002 list->item
Token=(268,ENT_0,ent)
Token=(339,EP_R,ep)
#081 register->EP
Token=(258,LABEL,L00003)
060 class2-operation->ent register INTLIT
014 operation->class3-operation
#008 statement->operation
#004 item->statement
#003 list->list item

Token=(270,RTN_0,rtn)
Token=(335,P_T,p)
#072 type->P
042 class1-operation->rtn type
rtn(04) p(07) 0(0000)
012 operation->class1-operation
#008 statement->operation
#004 item->statement
#003 list->list item

Token=(259,DEFINE,#define)
Token=(258,LABEL,L00002)
Token=(260,INTLIT,4)
#006 definition->DEFINE LABEL INTLIT
#005 item->definition
#003 list->list item

Token=(259,DEFINE,#define)
Token=(258,LABEL,L00003)
Token=(260,INTLIT,4)
#006 definition->DEFINE LABEL INTLIT
#005 item->definition
#003 list->list item
Token=(269,MST_0,mst)
Token=(260,INTLIT,0)
041 class1-operation->mst intlrit
mst(03) 0(00) 0(0000)
012 operation->class1-operation
#008 statement->operation
#004 item->statement
#003 list->list item
```

Figure 4. File p00.atrc

```
Token=(266,CUP_0,cup)
Token=(260,INTLIT,0)
Token=(258,LABEL,L00001)
059 class2-operation->cup INTLIT LABEL
014 operation->class3-operation
#008 statement->operation
#004 item->statement
#003 list->list item

Token=(314,STP_0,stp)
040 class0-operation->stp
stp(30) 0(00) 0(0000)
011 operation->class0-operation
#008 statement->operation
#004 item->statement
#003 list->list item

#001 program->list
```

Figure 4. File p00.atrc (continued)