The purpose of this lecture is to explain how P-Code instructions are translated to machine representation.

Please recall the machine formation for a P-Code instruction.

P-Code instructions consist of three fields, an operation code (opcode) followed by two operands. Operands depend on the opcode.

P-Code instructions occupy 32 bits or 4 bytes. The opcode is stored in the first byte of the instruction. The first operand, called *p*, is stored in the second byte and the second operand, called *q*, occupies the remaining two bytes as shown in Figure 7.

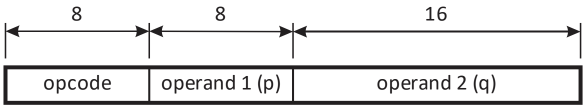

Figure 1. Anatomy of a P-Code instruction

The opcode and both operands are unsigned integers.

These instructions are stored in an array.

P-Code instructions are stored in an array. Each P-Code instruction occupies 4 bytes (32 bits). The number of P-Code instructions, n_p, is given by, n_p=l_p/4, where l_p is the value of the field labeled "P-Code instructions array length" in the directory.

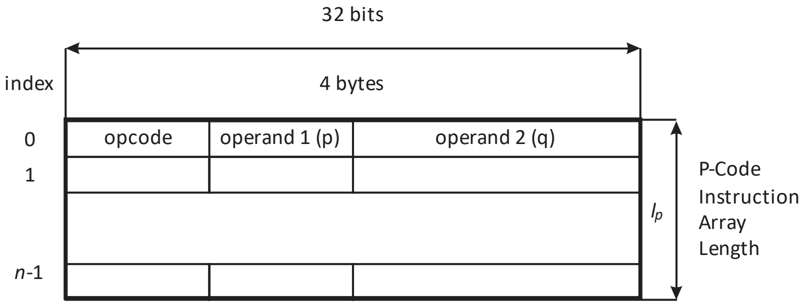

Figure 6. Anatomy of the P-Code instruction array

In this lecture, we focus on individual instructions, not how instructions are placed in the P-Code instruction array.

1. **cup** – Call User Procedure

P-Code Instruction			Machine Representation		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
cup	2	L00004	0x00	0x02	?

Notes:

- The mnemonic, cup, is translated to 0x00.
- Operand 1 is translated to hexadecimal – in this case – 0x02.
- Operand 2 is a label and not translated in this pass.

2. **csp** – Call Standard Procedure

P-Code Instruction			**Machine Representation**		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
csp		rdc	0x01	0x00	0x0001

Notes:

- The mnemonic, csp, is translated to 0x01.
- This instruction has no operand 1 and 0x00 is placed in the field for operand 1.
- Operand 2 is a standard function taken from Table 5 of the P-Machine Specification (https://cs2.uco.edu/~trt/cs5023/pspec.pdf). In this case, the standard function is read character that is translated to 0x0001.

3. **ent** – Entry (Subprogram prolog)

P-Code Instruction			**Machine Representation**		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
ent	sp	L00005	0x02	0x00	?

Notes:

- The mnemonic, ent, is translated to 0x02.
- Operand 1 is a register, either the stack pointer, sp, or the extreme pointer, ep. P-Machine register codes are given in Table 1 of the P-Machine Specification (https://cs2.uco.edu/~trt/cs5023/pspec.pdf).
- Operand 2 is a label and not translated in this pass.

4. **mst** – Mark Stack

P-Code Instruction			**Machine Representation**		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
mst		1	0x03	0x00	0x0001

Notes:

- The mnemonic, ent, is translated to 0x03.
- There is no operand 1 and 0x00 is placed in the field for operand 1.
- Operand 2 is a nonnegative integer value and translated to hexadecimal. In this case the value 1 is translated to 0x0001.

5. **rtn** – Return

P-Code Instruction			**Machine Representation**		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
rtn	p		0x04	0x07	0x0000

Notes:

- The mnemonic, rtn, is translated to 0x04.
- Operand 1 is a P-Machine type given in Table 2 of the P-Machine Specification (https://cs2.uco.edu/~trt/cs5023/pspec.pdf). In this case, the type p – procedure, is translated to 0x07.
- There is no operand 2 and 0x0000 is placed in the field for operand 2.

6. **equ** – Equality comparison

P-Code Instruction			Machine Representation		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
equ	i		0x05	0x03	0x0000

Notes:

- The mnemonic, equ, is translated to 0x05.
- Operand 1 is a P-Machine type given in Table 2 of the P-Machine Specification (https://cs2.uco.edu/~trt/cs5023/pspec.pdf). In this case, the type i – integer, is translated to 0x03.
- There is no operand 2 and 0x0000 is placed in the field for operand 2.

7. **neq** – Inequality comparison

P-Code Instruction			Machine Representation		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
neq	r		0x06	0x04	0x0000

Notes:

- The mnemonic, equ, is translated to 0x06.
- Operand 1 is a P-Machine type given in Table 2 of the P-Machine Specification (https://cs2.uco.edu/~trt/cs5023/pspec.pdf). In this case, the type r – real, is translated to 0x04.
- There is no operand 2 and 0x0000 is placed in the field for operand 2.

8. **grt** – Greater than comparison

P-Code Instruction			Machine Representation		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
grt	r		0x07	0x04	0x0000

Notes:

- The mnemonic, grt, is translated to 0x07.
- Operand 1 is a P-Machine type given in Table 2 of the P-Machine Specification (https://cs2.uco.edu/~trt/cs5023/pspec.pdf). In this case, the type r – real, is translated to 0x04.
- There is no operand 2 and 0x0000 is placed in the field for operand 2.

9. **geq** – Greater than or equal comparison

P-Code Instruction			Machine Representation		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
geq	r		0x08	0x04	0x0000

Notes:

- The mnemonic, geq, is translated to 0x08.
- Operand 1 is a P-Machine type given in Table 2 of the P-Machine Specification (https://cs2.uco.edu/~trt/cs5023/pspec.pdf). In this case, the type r – real, is translated to 0x04.
- There is no operand 2 and 0x0000 is placed in the field for operand 2.

10. **les** – Less than comparison

P-Code Instruction			Machine Representation		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
les	r		0x09	0x04	0x0000

Notes:

- The mnemonic, les, is translated to 0x09.
- Operand 1 is a P-Machine type given in Table 2 of the P-Machine Specification (https://cs2.uco.edu/~trt/cs5023/pspec.pdf). In this case, the type r – real, is translated to 0x04.
- There is no operand 2 and 0x0000 is placed in the field for operand 2.

11. **leq** – Less than or equal comparison

P-Code Instruction			Machine Representation		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
les	r		0x0A	0x04	0x0000

Notes:

- The mnemonic, leq, is translated to 0x0A.
- Operand 1 is a P-Machine type given in Table 2 of the P-Machine Specification (https://cs2.uco.edu/~trt/cs5023/pspec.pdf). In this case, the type r – real, is translated to 0x03.
- There is no operand 2 and 0x0000 is placed in the field for operand 2.

12. **adi** – integer addition

P-Code Instruction			Machine Representation		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
adi			0x0B	0x00	0x0000

Notes:

- The mnemonic, adi, is translated to 0x0B.
- There is no operand 1 and 0x00 is placed in the field for operand 1.
- There is no operand 2 and 0x0000 is placed in the field for operand 2.

13. **sbi** – integer subtraction

P-Code Instruction			Machine Representation		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
sbi			0x0C	0x00	0x0000

Notes:

- The mnemonic, sbi, is translated to 0x0C.
- There is no operand 1 and 0x00 is placed in the field for operand 1.
- There is no operand 2 and 0x0000 is placed in the field for operand 2.

14. **ngi** – integer sign inversion

P-Code Instruction			Machine Representation		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
ngi			0x0D	0x00	0x0000

Notes:

- The mnemonic, ngi, is translated to 0x0D.

- There is no operand 1 and 0x00 is placed in the field for operand 1.
- There is no operand 2 and 0x0000 is placed in the field for operand 2.

15. **mpi** – integer multiplication

P-Code Instruction			Machine Representation		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
mpi			0x0E	0x00	0x0000

Notes:

- The mnemonic, mpi, is translated to 0x0E.
- There is no operand 1 and 0x00 is placed in the field for operand 1.
- There is no operand 2 and 0x0000 is placed in the field for operand 2.

16. **dvi** – integer division

P-Code Instruction			Machine Representation		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
dvi			0x0F	0x00	0x0000

Notes:

- The mnemonic, dvi, is translated to 0x0F.
- There is no operand 1 and 0x00 is placed in the field for operand 1.
- There is no operand 2 and 0x0000 is placed in the field for operand 2.

17. **mod** – modulo

P-Code Instruction			Machine Representation		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
mod			0x10	0x00	0x0000

Notes:

- The mnemonic, mod, is translated to 0x10.
- There is no operand 1 and 0x00 is placed in the field for operand 1.
- There is no operand 2 and 0x0000 is placed in the field for operand 2.

18. **abi** – integer absolute value

P-Code Instruction			Machine Representation		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
abi			0x11	0x00	0x0000

Notes:

- The mnemonic, abi, is translated to 0x11.
- There is no operand 1 and 0x00 is placed in the field for operand 1.
- There is no operand 2 and 0x0000 is placed in the field for operand 2.

19. **sqi** – integer square

P-Code Instruction			Machine Representation		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
sqi			0x12	0x00	0x0000

Notes:

- The mnemonic, sqi, is translated to 0x12.
- There is no operand 1 and 0x00 is placed in the field for operand 1.

- There is no operand 2 and 0x0000 is placed in the field for operand 2.

20. **inc** – increment

P-Code Instruction			Machine Representation		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
inc	i		0x13	0x03	0x0000

Notes:

- The mnemonic, inc, is translated to 0x13.
- Operand 1 is a P-Machine type given in Table 2 of the P-Machine Specification (https://cs2.uco.edu/~trt/cs5023/pspec.pdf). In this case, the type i – integer, is translated to 0x03.
- There is no operand 2 and 0x0000 is placed in the field for operand 2.

21. **dec** – decrement

P-Code Instruction			Machine Representation		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
dec	i		0x14	0x03	0x0000

Notes:

- The mnemonic, dec, is translated to 0x14.
- Operand 1 is a P-Machine type given in Table 2 of the P-Machine Specification (https://cs2.uco.edu/~trt/cs5023/pspec.pdf). In this case, the type i – integer, is translated to 0x03.
- There is no operand 2 and 0x0000 is placed in the field for operand 2.

22. **adr** – real addition

P-Code Instruction			Machine Representation		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
adr			0x15	0x00	0x0000

Notes:

- The mnemonic, adr, is translated to 0x15.
- There is no operand 1 and 0x00 is placed in the field for operand 1.
- There is no operand 2 and 0x0000 is placed in the field for operand 2.

23. **sbr** – real subtraction

P-Code Instruction			Machine Representation		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
sbr			0x16	0x00	0x0000

Notes:

- The mnemonic, sbr, is translated to 0x16.
- There is no operand 1 and 0x00 is placed in the field for operand 1.
- There is no operand 2 and 0x0000 is placed in the field for operand 2.

24. **ngr** – real sign inversion

P-Code Instruction			Machine Representation		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
ngr			0x17	0x00	0x0000

Notes:

- The mnemonic, ngr, is translated to 0x17.
- There is no operand 1 and 0x00 is placed in the field for operand 1.
- There is no operand 2 and 0x0000 is placed in the field for operand 2.

25. **mpr** – real multiplication

P-Code Instruction			**Machine Representation**		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
mpr			0x18	0x00	0x0000

Notes:

- The mnemonic, mpr, is translated to 0x18.
- There is no operand 1 and 0x00 is placed in the field for operand 1.
- There is no operand 2 and 0x0000 is placed in the field for operand 2.

26. **dvr** – real division

P-Code Instruction			**Machine Representation**		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
dvr			0x19	0x00	0x0000

Notes:

- The mnemonic, dvr, is translated to 0x19.
- There is no operand 1 and 0x00 is placed in the field for operand 1.
- There is no operand 2 and 0x0000 is placed in the field for operand 2.

27. **abr** – real absolute value

P-Code Instruction			**Machine Representation**		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
abr			0x1A	0x00	0x0000

Notes:

- The mnemonic, abr, is translated to 0x1A.
- There is no operand 1 and 0x00 is placed in the field for operand 1.
- There is no operand 2 and 0x0000 is placed in the field for operand 2.

28. **sqr** – real square

P-Code Instruction			**Machine Representation**		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
sqr			0x1B	0x00	0x0000

Notes:

- The mnemonic, sqr, is translated to 0x1B.
- There is no operand 1 and 0x00 is placed in the field for operand 1.
- There is no operand 2 and 0x0000 is placed in the field for operand 2.

29. **ior** – inclusive OR

P-Code Instruction			**Machine Representation**		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
ior			0x1C	0x00	0x0000

Notes:

- The mnemonic, ior, is translated to 0x1C.
- There is no operand 1 and 0x00 is placed in the field for operand 1.
- There is no operand 2 and 0x0000 is placed in the field for operand 2.

30. **and** – AND

P-Code Instruction			**Machine Representation**		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
and			0x1D	0x00	0x0000

Notes:

- The mnemonic, and, is translated to 0x1D.
- There is no operand 1 and 0x00 is placed in the field for operand 1.
- There is no operand 2 and 0x0000 is placed in the field for operand 2.

31. **xor** – exclusive OR

P-Code Instruction			**Machine Representation**		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
xor			0x1E	0x00	0x0000

Notes:

- The mnemonic, xor, is translated to 0x1E.
- There is no operand 1 and 0x00 is placed in the field for operand 1.
- There is no operand 2 and 0x0000 is placed in the field for operand 2.

32. **not** – complement

P-Code Instruction			**Machine Representation**		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
not			0x1F	0x00	0x0000

Notes:

- The mnemonic, not, is translated to 0x1F.
- There is no operand 1 and 0x00 is placed in the field for operand 1.
- There is no operand 2 and 0x0000 is placed in the field for operand 2.

33. **inn** – set membership

P-Code Instruction			**Machine Representation**		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
inn			0x20	0x00	0x0000

Notes:

- The mnemonic, inn, is translated to 0x20.
- There is no operand 1 and 0x00 is placed in the field for operand 1.
- There is no operand 2 and 0x0000 is placed in the field for operand 2.

34. **uni** – set union

P-Code Instruction			**Machine Representation**		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
uni			0x21	0x00	0x0000

Notes:

- The mnemonic, uni, is translated to 0x21.

- There is no operand 1 and 0x00 is placed in the field for operand 1.
- There is no operand 2 and 0x0000 is placed in the field for operand 2.

35. **ntr** – set intersection

P-Code Instruction			Machine Representation		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
ntr			0x22	0x00	0x0000

Notes:

- The mnemonic, ntr, is translated to 0x22.
- There is no operand 1 and 0x00 is placed in the field for operand 1.
- There is no operand 2 and 0x0000 is placed in the field for operand 2.

36. **dif** – set difference

P-Code Instruction			Machine Representation		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
dif			0x23	0x00	0x0000

Notes:

- The mnemonic, dif, is translated to 0x23.
- There is no operand 1 and 0x00 is placed in the field for operand 1.
- There is no operand 2 and 0x0000 is placed in the field for operand 2.

37. **cmp** – set complement

P-Code Instruction			Machine Representation		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
cmp			0x24	0x00	0x0000

Notes:

- The mnemonic, cmp, is translated to 0x24.
- There is no operand 1 and 0x00 is placed in the field for operand 1.
- There is no operand 2 and 0x0000 is placed in the field for operand 2.

38. **sgs** – generate singleton set

P-Code Instruction			Machine Representation		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
sgs			0x25	0x00	0x0000

Notes:

- The mnemonic, sgs, is translated to 0x25.
- There is no operand 1 and 0x00 is placed in the field for operand 1.
- There is no operand 2 and 0x0000 is placed in the field for operand 2.

39. **ujp** – unconditional jump

P-Code Instruction			Machine Representation		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
ujp		L00004	0x26	0x00	0x0000

Notes:

- The mnemonic, ujp, is translated to 0x26.
- There is no operand 1 and 0x00 is placed in the field for operand 1.

- Operand 2 is a label. Labels are not addressed in this pass. The hexadecimal value 0x0000 is placed in operand 2.

40. **xjp** – indexed jump

P-Code Instruction			Machine Representation		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
ujp		L00004	0x27	0x00	0x0000

Notes:

- The mnemonic, xjp, is translated to 0x27.
- There is no operand 1 and 0x00 is placed in the field for operand 1.
- Operand 2 is a label. Labels are not addressed in this pass. The hexadecimal value 0x0000 is placed in operand 2.

41. **fjp** – false jump

P-Code Instruction			Machine Representation		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
fjp		L00004	0x28	0x00	0x0000

Notes:

- The mnemonic, fjp, is translated to 0x28.
- There is no operand 1 and 0x00 is placed in the field for operand 1.
- Operand 2 is a label. Labels are not addressed in this pass. The hexadecimal value 0x0000 is placed in operand 2.

42. **tjp** – true jump

P-Code Instruction			Machine Representation		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
tjp		L00004	0x29	0x00	0x0000

Notes:

- The mnemonic, tjp, is translated to 0x29.
- There is no operand 1 and 0x00 is placed in the field for operand 1.
- Operand 2 is a label. Labels are not addressed in this pass. The hexadecimal value 0x0000 is placed in operand 2.

43. **flt** – float top of stack

P-Code Instruction			Machine Representation		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
flt			0x2A	0x00	0x0000

Notes:

- The mnemonic, flt, is translated to 0x2A.
- There is no operand 1 and 0x00 is placed in the field for operand 1.
- There is no operand 2 and 0x0000 is placed in the field for operand 2.

44. **flo** – float next to top of stack

P-Code Instruction			Machine Representation		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
flo			0x2B	0x00	0x0000

Notes:

- The mnemonic, flo, is translated to 0x2B.
- There is no operand 1 and 0x00 is placed in the field for operand 1.
- There is no operand 2 and 0x0000 is placed in the field for operand 2.

45. **trc** – truncate

P-Code Instruction			**Machine Representation**		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
trc			0x2C	0x00	0x0000

Notes:

- The mnemonic, trc, is translated to 0x2C.
- There is no operand 1 and 0x00 is placed in the field for operand 1.
- There is no operand 2 and 0x0000 is placed in the field for operand 2.

46. **rnd** – round

P-Code Instruction			**Machine Representation**		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
rnd			0x2D	0x00	0x0000

Notes:

- The mnemonic, rnd, is translated to 0x2D.
- There is no operand 1 and 0x00 is placed in the field for operand 1.
- There is no operand 2 and 0x0000 is placed in the field for operand 2.

47. **chr** – convert to character

P-Code Instruction			**Machine Representation**		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
chr			0x2E	0x00	0x0000

Notes:

- The mnemonic, chr, is translated to 0x2E.
- There is no operand 1 and 0x00 is placed in the field for operand 1.
- There is no operand 2 and 0x0000 is placed in the field for operand 2.

48. **ord** – convert to integer

P-Code Instruction			**Machine Representation**		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
ord			0x2F	0x00	0x0000

Notes:

- The mnemonic, ord, is translated to 0x2F.
- There is no operand 1 and 0x00 is placed in the field for operand 1.
- There is no operand 2 and 0x0000 is placed in the field for operand 2.

49. **stp** – stop

P-Code Instruction			**Machine Representation**		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
stp			0x30	0x00	0x0000

Notes:

- The mnemonic, stp, is translated to 0x30.
- There is no operand 1 and 0x00 is placed in the field for operand 1.
- There is no operand 2 and 0x0000 is placed in the field for operand 2.

50. **lda** – load address of data

P-Code Instruction			Machine Representation		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
lda	0	5	0x31	0x00	0x0005

Notes:

- The mnemonic, lda, is translated to 0x31.
- Operand 1 is the number of static links to traverse to find the activation record for the referenced data. The value is converted to hexadecimal, 0x00.
- Operand 2 is the offset from the mark pointer where the reference data is stored. The value is converted to hexadecimal, 0x0005.

51. **ldc** – load constant

P-Code Instruction			Machine Representation		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
ldc	i	14	0x32	0x00	0x0000

Notes:

- The mnemonic, ldc, is translated to 0x32.
- Operand 1 is a P-Machine type given in Table 2 of the P-Machine Specification (https://cs2.uco.edu/~trt/cs5023/pspec.pdf). In this case, the type i – integer, is translated to 0x03.
- Operand 2 is the actual constant that must be either stored in a table or placed directly in operand 2. This instruction is not processed in this pass.

52. **ldi** – load indirect

P-Code Instruction			Machine Representation		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
ldi	0	0	0x33	0x00	0x0000

Notes:

- The mnemonic, ldi, is translated to 0x33.
- There is no operand 1 and 0x00 is placed in the field for operand 1.
- There is no operand 2 and 0x0000 is placed in the field for operand 2.

53. **lva** – load value (address)

P-Code Instruction			Machine Representation		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
lva	0	5	0x34	0x00	0x0005

Notes:

- The mnemonic, lda, is translated to 0x31.
- Operand 1 is the number of static links to traverse to find the activation record for the referenced data. The value is converted to hexadecimal, 0x00.
- Operand 2 is the offset from the mark pointer where the reference data is stored. The value is converted to hexadecimal, 0x0005.

54. **lvb** – load value (Boolean)

P-Code Instruction			Machine Representation		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
lvb	0	5	0x35	0x00	0x0005

Notes:

- The mnemonic, lvb, is translated to 0x35.
- Operand 1 is the number of static links to traverse to find the activation record for the referenced data. The value is converted to hexadecimal, 0x00.
- Operand 2 is the offset from the mark pointer where the reference data is stored. The value is converted to hexadecimal, 0x0005.

55. **lvc** – load value (character)

P-Code Instruction			Machine Representation		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
lvc	0	5	0x36	0x00	0x0005

Notes:

- The mnemonic, lvc, is translated to 0x36.
- Operand 1 is the number of static links to traverse to find the activation record for the referenced data. The value is converted to hexadecimal, 0x00.
- Operand 2 is the offset from the mark pointer where the reference data is stored. The value is converted to hexadecimal, 0x0005.

56. **lvi** – load value (integer)

P-Code Instruction			Machine Representation		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
lvi	0	5	0x37	0x00	0x0005

Notes:

- The mnemonic, lvi, is translated to 0x37.
- Operand 1 is the number of static links to traverse to find the activation record for the referenced data. The value is converted to hexadecimal, 0x00.
- Operand 2 is the offset from the mark pointer where the reference data is stored. The value is converted to hexadecimal, 0x0005.

57. **lvr** – load value (real)

P-Code Instruction			Machine Representation		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
lvr	0	5	0x38	0x00	0x0005

Notes:

- The mnemonic, lvr, is translated to 0x38.
- Operand 1 is the number of static links to traverse to find the activation record for the referenced data. The value is converted to hexadecimal, 0x00.
- Operand 2 is the offset from the mark pointer where the reference data is stored. The value is converted to hexadecimal, 0x0005.

58. **lvs** – load value (string)

P-Code Instruction			Machine Representation		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
lvs	0	5	0x39	0x00	0x0005

Notes:

- The mnemonic, lvs, is translated to 0x39.
- Operand 1 is the number of static links to traverse to find the activation record for the referenced data. The value is converted to hexadecimal, 0x00.
- Operand 2 is the offset from the mark pointer where the reference data is stored. The value is converted to hexadecimal, 0x0005.

59. **lvt** – load value (set)

P-Code Instruction			Machine Representation		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
lvs	0	5	0x3A	0x00	0x0005

Notes:

- The mnemonic, lvt, is translated to 0x3A.
- Operand 1 is the number of static links to traverse to find the activation record for the referenced data. The value is converted to hexadecimal, 0x00.
- Operand 2 is the offset from the mark pointer where the reference data is stored. The value is converted to hexadecimal, 0x0005.

60. **sti** – store indirect

P-Code Instruction			Machine Representation		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
sti	i	0	0x3B	0x03	0x0000

Notes:

- The mnemonic, sti, is translated to 0x3B.
- Operand 1 is a P-Machine type given in Table 2 of the P-Machine Specification (https://cs2.uco.edu/~trt/cs5023/pspec.pdf). In this case, the type i – integer, is translated to 0x03.
- There is no operand 2 and 0x0000 is placed in the field for operand 2.

61. **ixa** – compute indexed address

P-Code Instruction			Machine Representation		
mnemonic	**p**	**q**	**opcode**	**operand 1**	**operand 2**
ixa	0	4	0x3C	0x00	0x0004

Notes:

- The mnemonic, ixz, is translated to 0x3C.
- There is no operand 1 and 0x00 is placed in the field for operand 1.
- Operand 2 is the stride between adjacent elements in an array. Operand 2 is converted to hexadecimal.