**A: Lab 6.1**

3:

Toggle back and forth between *Symbolic* and *Binary* modes, until you can see the correspondence between the two representations of the same information. Then answer the following questions

1. At what address is the MUL X instruction stored? How is that address represented in binary?
   * MUL X = 4 = 00000100
2. What is the binary operation code that corresponds to the symbolic instruction LOD?
   * LOD = 00010100
3. How is the number 2 represented in binary?
   * 2 = 00000010
4. At what address is variable Y stored, and how is that address represented in binary?
   * Y = 10000010 = 130
5. Answer question d, this time for variable X.
   * X = 10000001 = 129
6. Answer question d, this time for variable W.
   * W = 10000000 = 126
7. What are the binary operation codes corresponding to the symbolic PIPPIN instructions ADD? MUL? STO?
   * They are the first half of the binary associated with the right column.
     + ADD = 00000000
     + MUL = 00000010
     + STO = 00000101

4:

What is the assembly language version program given in binary in part 4? Can you write the program (in part 4) a single equation? (For example, the program in part 1 can be written W = (2 + Y) \* X.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 0001 0100 | LOD | 0000 0101 | 5 |
| 2 | 0000 0000 | ADD | 1000 0000 | W |
| 4 | 0000 0001 | SUB | 1000 0011 | Z |
| 8 | 0000 0011 | DIV | 1000 0001 | X |
| 16 | 0000 0101 | STO | 1000 0010 | Y |
| 32 | 0000 1110 | NOP | 0000 0000 | 0 |
| 64 | 0000 1110 | NOP | 0000 0000 | 0 |
| 128 | 0000 1110 | NOP | 0000 0000 | 0 |

* + The above is written in Pippin. That is the assembly language used, or the symbolic language used rather.
  + Y = (5+W – Z) / X

**B: Lab 6.2**

7:

8:

9: