

Pass 2:

```

begin
  read first input line {from intermediate file}
  if OPCODE = 'START' then
    begin
      write listing line
      read next input line
    end {if START}
    write Header record to object program
    initialize first Text record
    while OPCODE ≠ 'END' do
      begin
        if this is not a comment line then
          begin
            search OPTAB for OPCODE
            if found then
              begin
                if there is a symbol in OPERAND field then
                  begin
                    search SYMTAB for OPERAND
                    if found then
                      store symbol value as operand address
                    else
                      begin
                        store 0 as operand address
                        set error flag (undefined symbol)
                      end
                end {if symbol}
              else
                store 0 as operand address
                assemble the object code instruction
              end {if opcode found}
            else if OPCODE = 'BYTE' or 'WORD' then
              convert constant to object code
            if object code will not fit into the current Text record then
              begin
                write Text record to object program
                initialize new Text record
              end
            end
            add object code to Text record
          end {if not comment}
          write listing line
          read next input line
        end {while not END}
        write last Text record to object program
        write End record to object program
        write last listing line
      end {Pass 2}

```

Figure 2.4(b) Algorithm for Pass 2 of assembler.

Line	Source statement
5	COPY START 0 COPY FILE FROM INPUT TO OUTPUT
10	FIRST STL RETADR SAVE RETURN ADDRESS
12	LDB #LENGTH ESTABLISH BASE REGISTER
13	BASE LENGTH
15	CLOOP +JSUB RDREC READ INPUT RECORD
20	LDA LENGTH TEST FOR EOF (LENGTH = 0)
25	COMP #0
30	JEQ ENDFIL EXIT IF EOF FOUND
35	+JSUB WRREC WRITE OUTPUT RECORD
40	J CLOOP LOOP
45	ENDFIL LDA EOF INSERT END OF FILE MARKER
50	STA BUFFER
55	LDA #3 SET LENGTH = 3
60	STA LENGTH
65	+JSUB WRREC WRITE EOF
70	J @RETADR RETURN TO CALLER
80	EOF BYTE C'EOF'
95	RETADR RESW 1
100	LENGTH RESW 1 LENGTH OF RECORD
105	BUFFER RESB 4096 4096-BYTE BUFFER AREA
110	.
115	.
120	.
125	RDREC CLEAR X CLEAR LOOP COUNTER
130	CLEAR A CLEAR A TO ZERO
132	CLEAR S CLEAR S TO ZERO
133	+LDT #4096
135	RLOOP TD INPUT TEST INPUT DEVICE
140	JEQ RLOOP LOOP UNTIL READY
145	RD INPUT READ CHARACTER INTO REGISTER A
150	COMPR A,S TEST FOR END OF RECORD (X'00')
155	JEQ EXIT EXIT LOOP IF EOR
160	STCH BUFFER,X STORE CHARACTER IN BUFFER
165	TIXR T LOOP UNLESS MAX LENGTH
170	JLT RLOOP HAS BEEN REACHED
175	EXIT STX LENGTH SAVE RECORD LENGTH
180	RSUB RETURN TO CALLER
185	INPUT BYTE X'F1' CODE FOR INPUT DEVICE
195	.
200	.
205	.
210	WRREC CLEAR X CLEAR LOOP COUNTER
212	LDT LENGTH
215	WLOOP TD OUTPUT TEST OUTPUT DEVICE
220	JEQ WLOOP LOOP UNTIL READY
225	LDCH BUFFER,X GET CHARACTER FROM BUFFER
230	WD OUTPUT WRITE CHARACTER
235	TIXR T LOOP UNTIL ALL CHARACTERS
240	JLT WLOOP HAVE BEEN WRITTEN
245	RSUB RETURN TO CALLER
250	OUTPUT BYTE X'05' CODE FOR OUTPUT DEVICE
255	END FIRST

Figure 2.5 Example of a SIC/XE program.