**Design Document Assignment 2**

**Purpose:**

The purpose of this program is to perform a first pass of the linker. The object is to scan through 2 or more XMO files and retrieve the publics (L1 records) alongside their addresses. S1 records are to be used to retrieve the starting addresses. A relocation address can be calculated with the S1 address and the number of data bytes encoded. This can determine the starting addresses of subsequent XMO modules.

**Algorithm:**

**Void Main Function (No Parameters):**

SET BOOLEAN FLAG valid

DO

PROMPT USER TO INPUT THE NUMBER OF XMO MODULES TO BE PARSED

STORE as count\_modules

IF INPUT IS NOT NUMERIC OR INPUT <=1

RUN AGAIN

END IF

END DO

WHILE FLAG != valid

END WHILE

DO

PROMPT USER TO INPUT NAME OF XMO FILE

STORE THE FILENAME IN FILENAME ARRAY

END DO

WHILE

COUNTER IS LESS THAN THE VARIABLE count\_modules

END WHILE

LOOP

PASS filename, current module counter, and max number of modules TO ParseFile function

END LOOP

END FUNCTION

**Void Parse File Function (Parameters:** filename, file counter, max number of files)

DECLARE last address in global

OPEN file

STORE contents of file in CHAR ARRAY wholefile

IF file counter==0

SET the last\_address variable to 0

END IF

DECLARE INT interim relocation address.

DECLARE INT publics variable

DECLARE iNT module starting address found as 0

LOOP through CHAR ARRAY wholefile

READ THROUGH IT LINE BY LINE

LOOP THROUGH EACH CHARACTER OF THE LINE

IF the first character is L && second character IS 1

SPLIT string array with “ “ delimiter

STORE the MIDDLE STRING AS THE symbol

STORE THE END STRING AS THE offset

SUM interim relocation address, offset, and last address

STORE in offsetted address variable

STORE symbol AND offsetted address in STRUCT TABLE

PRINT symbol, offset, and offsetted address

INCREMENT publics

END IF

ELSE IF the first character is S && second character is 1

IF THE module’s starting address found IS 0

LOOP through S1

STORE the 5th to 8th character in mod\_st\_address\_arr

CONVERT to int value mod\_starting\_address

SET mod\_starting\_address\_found to 1

END LOOP

END IF

LOOP THROUGH S1

STORE the 5th to 8th character in mod\_s1\_address\_arr

CONVERT using STRTOL to int value s1\_starting\_address

STORE the 2nd and 3rd character as character array length\_arr

CONVERT length\_arr to INTEGER length

CALCULATE num\_bytes = length-3 bytes (address and checksum)

STORE interim\_ra AS starting\_address + num\_bytes

END LOOP

END IF

IF publics ==0

PRINT “No Publics Found.”

END IF

STORE last\_address as interim\_ra+last\_address. (This will be stored globally to be accessible by the next file.)

RESET interim\_ra to 0

IF the current file is the last file

PRINT symbol table WITH symbol && address

END IF

END FUNCTION

**Data Dictionary:**

Table = 0 {Pubs}50 /\* Array of structs\*/

Pubs = (Pub\_Symbol + Pub\_Symbol\_Addr) /\* Struct of publics \*/

Pub\_Symbol\_Addr = Offsetted\_Addr

/\* Add the offset, the last address, and the relocation address within the module \*/

Offsetted\_Addr = Last\_Address + Interim\_RA + Offset

Module\_Starting\_Addr = Module\_Starting\_Addr + Last\_Address

/\*Relocation address within the module is called Interim\_RA\*/

Interim\_RA = S1\_Starting\_Address + Num\_Bytes

/\*Length is retrieved from the S1 record \*/

Num\_Bytes = Length – Addr\_Cksum\_Bytes

Addr\_Cksum\_Bytes = 3 /\*Address and checksum bytes = 3 total bytes\*/