ECED3403 – Assignment 1

Grace Yu

B00902046

May 15th, 2024

# Design

## Problem Introduction

This assignment aims to create a loader for an XM23p emulator. This loader will accept s-records obtained from XM23 assembly-language programs, before loading them into memory. These s-records contain instructions and data which will have their locations and values displayed to the user of the loader. Implementation of this loader is a crucial first step in fully developing the XM23p emulator.

## Design Section

Psuedocode:

MAIN:

IF no files added to executable THEN

PRINT “Please add file(s) or type input into console. \n”

ELSE IF file input is not valid THEN

PRINT “File failed to open. \n”

END IF

WHILE reading file input DO

SKIP first s0 entry

IF record is not s9 THEN

SAVE each s-record being read into 2D array

INCREMENT with each SAVE to count array size

END IF

END WHILE

WHILE

PROMPT user to specify IMEM or DMEM to display, as well as bounds

WHILE arrayplace is less than total array size DO

IF address field of s-record is within bounds DO

IF address field of s-record is larger than lower

bound DO

PRINT address field + “:”

FOR i index of lower bound to addressfield

PRINT “ 00”

SAVE ASCII character of 00 to asciiarray

INCREMENT row

IF row = 16 DO

PRINT new address field + “:”

PRINT asciiarray

PRINT “\n”

CONTINUE

END IF

END FOR

END IF

addressfieldend = addressfield + (2 \* length of

record)

FOR i index of addressfield to addressfieldend DO

PRINT data/instruction byte

SAVE ASCII character of data/instruction byte

to asciiarray

INCREMENT row

IF row = 16 DO

PRINT new address field + “:”

PRINT asciiarray

PRINT “\n”

CONTINUE

END IF

END FOR

END IF

END WHILE

IF currentaddressfield is less than upper bound DO

FOR i index of currentaddressfield to upper board DO

PRINT “ 00”

SAVE ASCII character of 00 to asciiarray

INCREMENT row

IF row = 16 DO

PRINT new address field + “:”

PRINT asciiarray

PRINT “\n”

CONTINUE

END IF

END FOR

END IF

## Data Dictionary

s-recordArray = 1{s-record} \* array that stores s-records \*

s-record = [s0 | s1 | s2 | s3]

s0 = “s0” + RecordLength + AddressField-s0 + SourceModuleName + CheckSum

RecordLength = 1{Byte}1 \* total length of the record in bytes \*

Byte = [0 – 9 | A – F | a – f] + [0 – 9 | A – F | a – f] \* bytes consist of hexadecimal values \*

AddressField-s0 = “0000” \* address ignored for s0 \*

SourceModuleName = 3{Byte}30 \* name of the source module, as ASCII \*

CheckSum = 1{Byte}1

s1 = “s1” + RecordLength + AddressField + DataInstruction + CheckSum

AddressField = 2{Byte}2 \* the memory location of the first byte \*

DataInstruction = 3{Byte}30 \* data/instruction bytes produced by the assembler from the original .ASM file \*

s2 = “s3” + RecordLength + AddressField + FirstDataInstrction + Checksum

FirstDataInstruction = 1{Byte}1 \* Only one byte produced because it comes from a BSS

asciiarray = 1{ASCII}

ASCII = [0 – 9 | A – F | a – f] + [0 – 9 | A – F | a – f]