SIC XE 2-pass C++ Assembler

◆ Team members:

- Ramy Wagdy Sobhy [no.21]
- Rofael Emil Fayez Behnam [no.22]
- Remon Hanna Wadie Youssef [no.23]
- Mohamed Ahmed Taher Mohamed Ahmed Elkholy [no.55]
- Mohamed Abd ElRahman ElFeki [no.59]

Requirements specifications

• implementing a 2-pass assembler for SIC XE Assembly, written in C/C++, producing code for the absolute loader used in the SIC programming assignments.

Specifications

- supports about **59** SIC XE architecture **instructions**.
- supports about 14 SIC XE architucture **pseudo-instructions** including [*EQU*, *ORG*, *LTORG*]
- supports about 10 registers.
- is executed through *assemble.cpp*.
- is to execute by entering "assemble <source-file-name>" in OS shell.
- supports *simple expression evaluation* including operands [+, -, *,/]
- supports **Literals** (Including *LTORG*) in the forms [=*C'*<*ASCII-TEXT*>', =*X'HEX-TEXT*'] [**Bonus Feature**]
- supports Control sections [Bonus Feature]

- Design
- 9 *C/C*++ code files
 - "assemble.cpp" [contains main function only]
 - "SICXEPass1.cpp" with header file "SICXEPass1.h" [performs Pass1 algorithm]
 - Input: input.txt
 - Output: *Intermediate_File.txt*
 - Listing File Array will be implemented as following:

listingFile[0] -> Line Number

listingFile[1] -> Location Counter

listingFile[2] -> Label If found

listingFile[3] -> Operation Code

listingFile[4] -> Flags

listingFile[5] -> Operands

listingFile[6] -> Comments _If found_

listingFile[7] -> Object Code for pass2

listingFile[8] -> Modification Boolean for pass2

- vector<string> starts; [address of start of each control section]
- vector<string> *ends*; [address of **end of each control section**]
- bool pass1(); [performs pass1 algorithm stated by Leland L. Beck]
- void generateIntermediateFile(); [write to **O/P file**]
- string *evaluateExpression*(string expression);
- bool isComment(string line);
- int *verifyLine*(string line); [checks **validity of code** statements]
- vector<string> split(string s); [parses code statements recursively]
- bool isLiteralFoundListingFile(int lineNum, string literal);
- vector<string> splitOperators(string expression);
- "SICXEPass2.cpp" with its header file "SICXEPass2.h" [pass2 implementation]
 - Input: ListingFile array generated by SICXEPass1.cpp
 - Output: Intermediate_File.txt, Object.o
- void getObjectCode(int line); [generates Object Code of each statement]
- void setFlags(string operand,int line); [sets **n,i, x, b, p, e** instruction bits]
- bool *pass2*(); [performs **pass2 algorithm** stated by Leland L. Beck]
- void *flushRecord*(); [writes to **O/P file**]
- bool *headerRecord*(string programName, string startAdd, string endAdd); [generates **H records**]
- bool flushTextRecord(); [writes T record to O/P file]
- bool *appendTextRecord*(string locationCounter, string objectCode, bool executable); [generates **T records**]

- bool *endRecord*(); [generates **E records**]
- bool *modificationRecord*(string address, string length, string details); [generates **M records**]
- bool *addExternalDefinition*(string name, string address); [generates **D records**]
- bool *addExternalReference*(string name); [generates **R record**]
- "Tables.cpp" associated with "Tables.h" [contains main data structures]
- "Conversions.cpp" associated with "Conversions.h" [contains ASCII-number-system conversions]
 - supports ASCII, Binary, Decimal and Hexadecimal 2-way conversions.

Main data structures

- Hash Table
- ◆ **SYMTAB:** map<string, vector<string> > symTab;
- **Registers:** map<string, int> registersTable;
- ◆ **OPTAB:** map<string, string> opTab;
 - ◆ Set
- pseudo-instructions: set<string> directivesSet;
 - ◆ 2-dimensional array
- ◆ **LITTAB**: string LITTAB[1000][4];

♦ Algorithms description

[O(N), N is the count of ASCII characters in I/O file]

Pass1

```
begin
  read first input line
  if OPCODE = 'START' then
      begin
         save #[OPERAND] as starting address
         initialize LOCCTR to starting address
         write line to intermediate file
         read next input line
      end (if START)
  else
      initialize LOCCTR to 0
  while OPCODE # 'END' do
      begin
         if this is not a comment line then
             begin
                if there is a symbol in the LABEL field then
                   begin
                       search SYMTAB for LABEL
                       if found then
                          set error flag (duplicate symbol)
                           insert (LABEL, LOCCTR) into SYMTAB
                   end (if symbol)
                search OPTAB for OPCODE
                if found then
                   add 3 (instruction length) to LOCCTR
                else if OPCODE = 'WORD' then
                   add 3 to LOCCTR
                else if OPCODE = 'RESW' then
                   add 3 * #[OPERAND] to LOCCTR
                else if OPCODE = 'RESR' then
                   add #[OPERAND] to LOCCTR
                else if OPCODE = 'BYTE' then
                   begin
                       find length of constant in bytes
                       add length to LOCCTR
                   end (if BYTE)
                else
                   set error flag (invalid operation code)
             end (if not a comment)
         write line to intermediate file
         read next input line
     end (while not END)
  write last line to intermediate file
  save (LOCCTR - starting address) as program length
end (Pass 1)
```

```
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 than
                          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
                 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)
```

◆ Assumptions

- Free-formated I/O.
- Memory of target machine does not exceed 1 MB.
- Hexadecimal literals must fill completely one or multiple byte-blocks.

♦ TeamWord distribution

- Ramy Wagdy Sobhy
 - Code statements structure verification [pass1]
 - geberating object code for each code statement [pass2]
- Rofael Emil Fayez Behnam
 - Free-format recursive parsing [pass1]
 - generating Object File Records [pass2]
- Remon Hanna Wadie Youssef
 - generating SYMTAB [pass1]
 - parsing EXTREF and EXTDEF operands [pass2]
 - QT-creator GUI with syntax highlighting feature
- Mohamed Ahmed Taher Mohamed Ahmed Elkholy
 - pass2 main algorithm [pass2]
 - QT-creator GUI with auto-compeletion feature
- Mohamed Abd ElRahman ElFeki
 - pass1 main algorithm [pass1]
 - generating LITAB
 - construction OPTAB [pre-design]

```
.dadssa
              START
2
    COPY
3
              EXTDEF
                      BUFFER,
                                 BUFEND, LENGTH
 4
              EXTREF
                      RDREC,
                                      WRREC
    FIRST
                       RETADR
 6
    CL00P
              +JSUB
                       RDREC
              LDA
                       LENGTH, X
                                  .hello World
              COMP
 8
                       #0
                       ENDFIL
              JEQ
9
              +JSUB
                      WRREC
10
                       CL00P
11
                          C'EO
                                           F'
    ENDFIL
              LDA
12
13
              STA
                       BUFFER
14
              LDA
15
              STA
                       LENGTH
                                WRREC!
                        JSUB
16
                       0
                                 RETADR
17
    RETADR
              RESW
18
    LENGTH
19
              RESW
                       1
20
              LTORG
    BUFFER
                       4096
21
              RESB
22
    BUFEND
              EQU
23
    MAXLEN
              EQU
                      BUFEND
                                                BUFFER
24
    RDREC
              CSECT
25
              EXTREF
                      BUFFER, LENGTH, BUFEND
              CLEAR
26
27
              CLEAR
              CLEAR
28
29
              +LDT
                       #MAXLEN2
30
    RL00P
              TD
                       INPUT
31
              JEQ
                       RLOOP
32
              RD
                       INPUT
              COMPR
                      A,S
33
                       EXIT
34
              JEQ
              +STCH
                       BUFFER,X
35
36
              TIXR
                      RL00P
37
              JLT
38
    EXIT
              +STX
                       LENGTH
              RSUB
40
    INPUT
              BYTE
41
    MAXLEN2
              WORD
                      BUFEND-BUFFER
42
    WRREC
               CSECT
43
                      LENGTH, BUFFER
              EXTREF
44
              CLEAR
                    LENGTH
45
            +LDT
    WLOOP
                       =X'05'
46
              TD
47
              JEQ
                       WLOOP
              +LDCH
                       BUFFER, X
49
              WD
                       =X'05'
              TIXR
50
                      WLOOP
51
              JLT
              RSUB
52
                       FIRST
              END
53
```

1	0		.dadssa					
2	1	0000	COPY	START	0			
3	2	0000		EXTDEF	BUFFER, BUFEND, LENGTH			
4	3			EXTREF	RDREC, WRREC			
5	4	0000	FIRST	STL	RETADR			
6	5	0003	CL00P	+JSUB	RDREC			
7	6	0007		LDA	LENGTH,X	.h	ello World	
8	7	A000		COMP	#0			
9	8	000D		JEQ	ENDFIL			
10 11	10	0010 0014		+JSUB L	WRREC CLOOP			
12	11	0017	ENDFIL	LDA	=C'EO F'			
13	12	001A	LIND! IL	STA	BUFFER			
14	13	001D		LDA	#3			
15	14	0020		STA	LENGTH			
16	15	0023	+	JSUB				
17	16		**** Error: In					
18	17	0023	DETADD	7	@RETADR			
19	18 19	0026 0029	RETADR	RESW	1			
20 21	20	0029	LENGTH	RESW LTORG	'			
22	21	002C	*	=C'E0	F'			
23	22	003A	BUFFER	RESB	4096			
24	23	0103	BUFEND	EQU	*			
25	24	0100	MAXLEN	EQU	BUFEND-BUFFER			
26	25	0000	RDREC	CSECT				
27	26			EXTREF	BUFFER, LENGTH, BUFEND			
28	27	0000		CLEAR	×			
29	28	0002		CLEAR	A			
30 31	29 30	0004 0006		CLEAR +LDT	S #MAXLEN2			
32	31	0000 000A	RLOOP	TD	INPUT			
33	32	000D	KLOOI	JEQ	RLOOP			
34	33	0010		RD	INPUT			
35	34	0013		COMPR	A,S			
36	35	0015		JEQ	EXIT			
37	36	0018		+STCH	BUFFER, X			
38	37	001C		TIXR	T			
39 40	38 39	001E 0021	EXIT	JLT +STX	RLOOP LENGTH			
41	40	0021	EVII	RSUB	LENGIN			
42	41	0028	INPUT	BYTE	X'F1'			
43	42	0029	MAXLEN2	WORD	BUFEND-BUFFER			
44	43	0000	WRREC	CSECT				
45	44			EXTREF	LENGTH, BUFFER			
46	45	0000		CLEAR	X			
47	46	0002	VII 00D	+LDT	LENGTH			
48 49	47 48	0006 0009	WLOOP	JEQ	=X ' 05 ' WLOOP			
50	49	000G		+LDCH	BUFFER, X			
51	50	0010		WD	=X'05'			
52	51	0013		TIXR	Т			
53	52	0015		JLT	WLOOP			
54	53	0018		RSUB				
55	54	001B		END	FIRST			
56	55	001B	*	=X'05'				
57								
58 59			Literals TABLE					
60	***	****	LILEIAIS IADLE	******	**			
61	*	Name	Value Length	Address	*			
62		=C'E0	F'		0000000000000000000046	14	002C	
63		=X ' 05 '		001B				
64			******					
65	***	****	******	*******	***			
66 67								
n/								

51		0010 0013		WD		=X * 05 * T					
52 53	52	0015		TIXR JLT		WLOOP					
54 55		0018 001B		RSUB END		FIRST					
56		001B	*	=X'05'		FIRST					
57											
58 59			iterals TABL								
60	*******		**********								
61 62		Name V =C'EO	F'	h Address 454F000	* 0000000000000	000000046	14 002C				
63		=X'05' 0	5 2	001B							
64 65			*********								
66											
67 68		SYMBOL TA	BLE								
69	******	*******	******	******							
70 71	* SYME		ADDRESS	*Sectio							
72	*BUFEND	!	103A	0	*						
73 74	*BUFFER *CLOOP	- 1	003A 0003	0							
75	*COPY	į	0000	0	*						
76 77	*ENDFIL	- 1	0017 0021	0							
78	*FIRST	į	0000	0	*						
79 80	*INPUT *LENGTH	- 1	0028 0029	1 0							
81	*MAXLEN	. !	103A	0	*						
82 83	*MAXLEN2 *RDREC	١	0029 103A	1	:						
84	*RETADR	į	0026	0	*						
85 86	*RLOOP *WLOOP	- 1	000A 0006	1 2	:						
87	*WRREC		002C	2	*						
88 89			r SIC/XE Ass		******						
90	Line	LocCtr	Labels	0pCode	Flags	Opera	ands Com	nments	ObjectCode	ModBoolean	
91 92	0	000	dadssa 0	COPY	START		0				
93	2	000			EXTDEF		BUFFER, BUFEND, LE	NGTH			-
94 95	3 4	000	0 F	IRST	EXTREF STL	110010	RDREC, WRREC RETADR			172023	
96	5	000	3 0	LOOP	+JSUB	110001	RDREC	h-11- 1		B10103A	
97 98	6 7	000			LDA COMP	111010 011000	LENGTH,X #0	.hello \	World	03A01F 298000	
99	8	000	D		JEQ	110010	ENDFIL			332007	
100	,	001	0		+JSUB	110001	WRREC		-	B10002C	
101	10	0	014		J	110010	CL00P			3F2FEC	
102	11		017	ENDFIL	LDA	110010					022012
103 104	12	U			CTA	110010	=C,EO	F'		053010	032012
	13	0	01A 01D		STA LDA	110010 010000	BUFFER #3	F.		0F201D 010003	032012
105	14	0			LDA STA	110010	BUFFER	F.		0F201D 010003 0F2006	032012
106	14 15		01D 020 +		LDA STA SUB WRREC!	110010 010000	BUFFER #3	F.		010003	032012
106 107 108	14 15 16 17	0023	01D 020 **** Error: 023	Invalid (LDA STA SUB WRREC! Operand J	110010 010000	BUFFER #3 LENGTH @RETADR	F.		010003 0F2006 3E2000	032012
106 107 108 109	14 15 16 17 18	0023	01D 020 **** Error: 023 026	Invalid (LDA STA SUB WRREC! Operand J RESW	110010 010000 110010	BUFFER #3 LENGTH @RETADR 1	F.		010003 0F2006 3E2000	032012
106 107 108 109 110 111	14 15 16 17 18 19 20	0023	01D 020 **** Error: 023 026 029	Invalid (RETADR LENGTH	LDA STA SUB WRREC! Operand J RESW RESW LTORG	110010 010000 110010 100010	BUFFER #3 LENGTH @RETADR	F.		010003 0F2006 3E2000	032012
106 107 108 109 110 111 112	14 15 16 17 18 19 20 21	0023	01D 020 + **** Error: 023 024 026 029	Invalid (RETADR LENGTH	LDA STA SUB WRREC! Operand J RESW RESW LTORG =C'EO	110010 010000 110010	BUFFER #3 LENGTH @RETADR 1	,		010003 0F2006 3E2000	032012
106 107 108 109 110 111	14 15 16 17 18 19 20	0023	01D 020 **** Error: 023 026 029	Invalid (RETADR LENGTH	LDA STA SUB WRREC! Operand J RESW RESW LTORG	110010 010000 110010 100010	BUFFER #3 LENGTH @RETADR 1	ř.		010003 0F2006 3E2000	032012
106 107 108 109 110 111 112 113 114 115	14 15 16 17 18 19 20 21 22 23 24	0023 00 00 00 00 00 00 00	01D 020 + **** Error: 023 026 029 02C 03A 103	RETADR LENGTH * BUFFER BUFFER BUFEND MAXLEN	LDA STA SUB WRREC! Operand J RESW RESW LTORG =C'EO RESB EQU EQU	110010 010000 110010 100010	BUFFER #3 LENGTH @RETADR 1 1			3E2000	032012
106 107 108 109 110 111 112 113 114	14 15 16 17 18 19 20 21 22 23	0023 00 00 00 00 00 00 00	01D 020 + **** Error: 023 026 029 02C 03A 103	RETADR LENGTH * BUFFER BUFFER BUFEND	LDA STA SUB WRREC! Operand J RESW RESW LTORG =C'EO RESB EQU	110010 010000 110010 100010	BUFFER #3 LENGTH @RETADR 1 1	ER		010003 0F2006 3E2000	
106 107 108 109 110 111 112 113 114 115 116 117 118	14 15 16 17 18 19 20 21 22 23 24 25 26 27	0023 0000 000 000 000 000 000 000	01D 020 + **** Error: 023 026 029 02C 03A 103 100 000	RETADR LENGTH * BUFFER BUFFER BUFEND MAXLEN	LDA STA STA WRREC! Operand J RESW LTORG =C'EO RESB EQU EQU CSECT EXTREF CLEAR	110010 010000 110010 100010	BUFFER #3 LENGTH @RETADR 1 1 4096 * BUFFEND-BUFFI BUFFER, LENG	ER		010003 0F2006 3E2000 	
106 107 108 109 110 111 112 113 114 115 116 117 118	14 15 16 17 18 19 20 21 22 23 24 25 26 27 28	0023 00 00 00 00 00 00 00 00 00 00	01D 020 020 023 026 029 02C 03A 103 100 000	RETADR LENGTH * BUFFER BUFFER BUFEND MAXLEN	LDA STA STA SUB WRRC! Operand RESW RESW LTORG =C'EO RESB EQU CSECT EXTREF CLEAR CLEAR	110010 010000 110010 100010	BUFFER #3 LENGTH @RETADR 1 1 4096 ** BUFFEND-BUFFE BUFFER, LENG X A	ER		010003 0F2006 3E2000 B410 B400	
106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121	14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30	0023 00 00 00 00 00 00 00 00 00 00 00 00	01D	RETADR LENGTH * BUFFER BUFEND MAXLEN RDREC	LDA STA STA WRREC! Operand J RESW LTORG =C'EO RESB EQU EQU CSECT EXTREF CLEAR CLEAR CLEAR +LDT	110010 010000 110010 100010 110000 F'	BUFFER #3 LENGTH @RETADR 1 1 4096 * BUFFEND-BUFFI BUFFER, LENG X A S #MAXLEN2	ER		010003 0F2006 3E2000 B410 B400 B440 75100029	
106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122	14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31	0023 00 00 00 00 00 00 00 00 00 00 00 00 00	01D	RETADR LENGTH * BUFFER BUFFER BUFEND MAXLEN	LDA STA STA STA STA Perand J RESW RESW LTORG =C'EO RESB EQU CSECT EXTREF CLEAR CLEAR CLEAR TD	110010 010000 110010 100010 110000 F' 110000	BUFFER #3 LENGTH @RETADR 1 1 4096 * BUFEND-BUFF BUFFER, LENG X A S #MAXLENZ INPUT	ER		010003 0F2006 3E2000 B410 B400 B440 75100029 E32018	
106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124	14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33	0023 00 00 00 00 00 00 00 00 00 00 00 00 00	01D 020 + **** Error: 023 026 029 022C 033A 103 100 000 0000 0000 0002 0004 0006 000A 000B 000D 000D	RETADR LENGTH * BUFFER BUFEND MAXLEN RDREC	LDA STA STA STA STA STA STA STA STA STA ST	110010 010000 110010 100010 110000 F'	BUFFER #3 LENGTH @RETADR 1 1 4096 * BUFEND-BUFFI BUFFER, LENG X A S #MAXLEN2 INPUT RLOOP INPUT	ER		010003 0F2006 3E2000 B410 B400 B440 75100029 E32018 332FFA DB2015	
106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125	14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34	0023 00 00 00 00 00 00 00 00 00 00 00 00 00	#### Error: 0229 0220 024 0229 02C 03A 103 103 100 000 000 000 000 000 000 000	RETADR LENGTH * BUFFER BUFEND MAXLEN RDREC	LDA STA STA STA STA STA STA STA RESW RESW LTORG =C'EO RESB EQU EQU CSECT EXTREF CLEAR CLEA	110010 010000 110010 100010 110000 F. 110000 010001 110010 110010 110010	BUFFER #3 LENGTH @RETADR 1 1 4096 * BUFFEND-BUFFI BUFFER, LENG X A S #MAXLENZ INPUT RLOOP INPUT A,S	ER		010003 0F2006 3E2000 B410 B4400 B4407 5100029 E3201B 332FFA DB2015 A004	
106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124	14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 30 31 32 33 34 35 36	0023 00 00 00 00 00 00 00 00 00 00 00 00 00	01D 020 + **** Error: 023 026 0229 022C 033A 103 100 000 000 0000 0000 0000 000	RETADR LENGTH * BUFFER BUFEND MAXLEN RDREC	LDA STA STA STA STA STA STA STA STA STA ST	110010 010000 110010 100010 110000 F' 110000 010001 110010	BUFFER #3 LENGTH @RETADR 1 1 4096 * BUFEND-BUFFI BUFFER, LENG X A S #MAXLEN2 INPUT RLOOP INPUT	ER		010003 0F2006 3E2000 B410 B400 B440 75100029 E32018 332FFA DB2015	
106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128	14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 37	0023 00 00 00 00 00 00 00 00 00 00 00 00 00	#### Error: 022 024 022 026 029 02C 03A 103 103 000 000 000 000 000 000 000 000	RETADR LENGTH * BUFFER BUFEND MAXLEN RDREC	LDA STA STA SUB WRREC! Operand RESW RESW LTORG =C'EO RESB EQU CSECT EXTREF CLEAR CL	110010 010000 110010 100010 110000 F. 110000 010001 110010 110010 110010 110010	BUFFER #3 LENGTH @RETADR 1 1 4096 * BUFFEND-BUFFI BUFFER, LENG X A S #MAXLENZ INPUT RLOOP INPUT A,S EXIT BUFFER, X	ER		010003 0F2006 3E2000 B410 B4400 B4407 5100029 E3201B 332FFA DB2015 A004 332009 5790003A B850	
106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127	14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39	0023 00 00 00 00 00 00 00 00 00 00 00 00 00	01D 020 + **** Error: 023 026 0229 022C 033A 103 100 000 000 0000 0000 0000 000	RETADR LENGTH * BUFFER BUFEND MAXLEN RDREC	LDA STA STA STA STA STA STA STA STA STA ST	110010 010000 110010 100010 110000 F' 110000 010001 110010 110010 110010	BUFFER #3 LENGTH @RETADR 1 1 4096 * BUFEND-BUFFI BUFFER, LENG X A S #MAXLEN2 INPUT RLOOP INPUT A, S EXIT BUFFER, X	ER		010003 0F2006 3E2000 B410 B400 B440 75100029 E3201B 332FFA DB2015 A004 332009 5790003A B850 3B2FE9 13100029	
106 107 108 110 111 112 113 114 115 116 117 118 120 121 122 123 124 125 126 127 128 129 130 131	14 15 16 17 18 19 20 21 22 23 24 25 27 28 29 30 31 32 33 34 35 36 37 38 39 40	0023 00 00 00 00 00 00 00 00 00 00 00 00 00	#### Error: 022 024 027 028 029 020 020 020 020 020 020	RETADR LENGTH * BUFFER BUFFER BUFEND MAXLEN RDREC	LDA STA STA STA STA STA STA STA RESW RESW LTORG =C'EO RESB EQU CSECT EXTREF CLEAR CLEAR CLEAR CLEAR CLEAR CLEAR CLEAR TD JEQ STC TD JEQ COMPR JEQ STC TIX STC TIX STC TIX STX RSUB	110010 010000 110010 100010 110000 F' 110000 010001 110010 110010 110010 110010 110010	BUFFER #3 LENGTH @RETADR 1 1 4096 * BUFFEND-BUFFI BUFFER, LENGT X A S #MAXLENZ INPUT RLOOP INPUT A,S EXIT BUFFER,X T RLOOP LENGTH	ER		010003 0F2006 3E2000 B410 B4400 B4407 5100029 E32018 332FFA DB2015 A004 332009 5790003A B850 3B2FE9 13100029 4F0000	
106 107 108 110 111 112 113 114 115 116 117 118 120 121 122 123 124 125 126 127 128 129 130	14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39	0023 0000 000 000 000 000 000 000 000 00	#### Error: 0229 02C 033A 103 100 000 000 000 000 000 000 000 000	RETADR LENGTH * BUFFER BUFEND MAXLEN RDREC	LDA STA STA STA STA STA STA STA STA STA ST	110010 010000 110010 100010 110000 F' 110000 010001 110010 110010 110010 110010 110010 110010 110010	BUFFER #3 LENGTH @RETADR 1 1 4096 BUFEND-BUFF BUFFER, LENG X A S #MAXLENZ INPUT RLOOP INPUT A, S EXIT BUFFER, X T RLOOP	ER TH, BUFEND		010003 0F2006 3E2000 B410 B400 B440 75100029 E3201B 332FFA DB2015 A004 332009 5790003A B850 3B2FE9 13100029	
106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 133 134	14 15 16 17 18 19 20 21 22 23 24 25 27 28 29 30 31 32 33 34 35 36 37 38 40 41 42 43	0023 00 00 00 00 00 00 00 00 00 00 00 00 00	#### Error: 0229 02C 033A 103 100 000 000 000 000 000 000 000 000	RETADR LENGTH * BUFFER BUFEND MAXLEN RDREC RLOOP	LDA STA STA STA STA STA STA STA RESW RESW LTORG =C'EO RESB EQU CSECT EXTREF CLEAR CLEAR CLEAR CLEAR CLEAR CLEAR TD JEQ STC FT RD COMPR JEQ STC	110010 010000 110010 100010 110000 F' 110000 010001 110010 110010 110010 110010 110010 110010 110010	BUFFER #3 LENGTH @RETADR 1 1 4096 * BUFEND-BUFFI BUFFER, LENGT X A S #MAXLEN2 INPUT RLOOP INPUT A,S EXIT BUFFER,X T RLOOP LENGTH X'F1' BUFEND-BUFFI	ER TH, BUFEND		010003 0F2006 3E2000 B410 B4400 B4400 B4407 5100029 E3201B 332FFA DB2015 A004 332009 5790003A B850 3B2FE9 13100029 4F0000 F1	
106 107 108 109 110 111 112 113 114 115 116 117 118 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135	14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44	0023 00 00 00 00 00 00 00 00 00 00 00 00 00	#### Error: 0229 02C 033A 103 100 000 000 000 000 000 000 000 000	RETADR LENGTH * BUFFER BUFEND MAXLEN RDREC RLOOP EXIT INPUT MAXLEN2	LDA STA STA STA STA STA STA STA STA STA ST	110010 010000 110010 100010 110000 F' 110000 010001 110010 110010 110010 110010 110010 110010 110010 110010 110010	BUFFER #3 LENGTH @RETADR 1 1 4096 BUFEND-BUFFI BUFFER, LENG X A S #MAXLENZ INPUT RLOOP INPUT A, S EXIT BUFFER, X T RLOOP LENGTH X'F1' BUFEND-BUFFI LENGTH, BUFFI	ER TH, BUFEND		010003 0F2006 3E2000 B410 B4400 B4400 75100029 E3201B 332FFA DB2015 A004 332009 5790003A 382FE9 13100029 4F0000 F1	
106 107 108 109 110 111 112 113 114 115 116 117 118 120 121 122 123 124 125 127 128 129 130 131 131 133 134 135 136 137	14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 44 44 45 46	0023 00 00 00 00 00 00 00 00 00 00 00 00 00	#### Error: 0220 #### Error: 0226 0229 02C 03A 103 103 100 000 000 000 000	RETADR LENGTH * BUFFER BUFFER BUFEND MAXLEN RDREC RLOOP EXIT INPUT MAXLENZ WRREC	LDA STA STA STA STA STA STA STA RESW RESW LTORG =C'EO RESB EQU CSECT EXTREF CLEAR CLEAR CLEAR CLEAR CLEAR CLEAR TD JEQ STCH STCH TTD JEQ COMPR JEQ STCH STCH TTXR JLT +STCH TXR SUB BYTE WORD CSECT EXTREF CLEAR LDT LST LST LST LST LST LST LST LST LST LS	110010 010000 110010 100010 110000 110000 110000 110010 110010 110010 110010 110000 110000	BUFFER #3 LENGTH @RETADR 1 1 4096 * BUFEND-BUFFI BUFFER, LENGY X A S #MAXLEN2 INPUT RLOOP INPUT A,S EXIT BUFFER, X T RLOOP LENGTH X'F1' BUFEND-BUFFI LENGTH, BUFFI X LENGTH, BUFFI X LENGTH	ER TH, BUFEND		010003 0F2006 3E2000 B410 B4400 B4400 B4400 B4400 B4400 5100029 E3201B 332FFA DB2015 A004 332009 5790003A B850 3B2FE9 13100029 4F0000 F1 B410 77100029	
106 107 108 109 110 111 112 113 114 115 116 117 120 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 137 138	14 15 16 17 18 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47	0023 0000 000 000 000 000 000 000 000 00	#### Error: 022 023 024 029 02C 033A 103 103 100 000 000 000 0000 0010 0113 0115 0118 011C 011E 021 022 028 0229 000 000 0000 0000 0000 000	RETADR LENGTH * BUFFER BUFEND MAXLEN RDREC RLOOP EXIT INPUT MAXLEN2	LDA STA STA STA STA STA STA STA STA STA ST	110010 010000 110010 100010 110000 110000 110000 110010 110010 110010 110010 110000 110000	BUFFER #3 LENGTH @RETADR 1 1 4096 * BUFFEND-BUFFI BUFFER, LENGY A S #MAXLENZ INPUT RLOOP INPUT A,S EXIT BUFFER, X T RLOOP LENGTH X'F1' BUFEND-BUFFI LENGTH, BUFFI X LENGTH, BUFFI X LENGTH	ER TH, BUFEND		010003 0F2006 3E2000 B410 B4400 B4400 B4400 75100029 E3201B 332FFA DB2015 A004 332009 5790003A B850 3B2FE9 13100029 4F0000 F1 B410 77100029 E32012	
106 107 108 109 110 111 112 113 114 115 116 117 118 120 121 122 123 124 125 127 128 129 130 131 131 133 134 135 136 137	14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 30 31 33 34 35 36 37 38 39 40 41 42 43 44 44 45 46 47 48 49 49 49 49 49 49 49 49 49 49 49 49 49	0023 00 00 00 00 00 00 00 00 00 00 00 00 00	#### Error: 0220 #### Error: 0226 0229 02C 03A 103 103 100 000 000 000 000	RETADR LENGTH * BUFFER BUFFER BUFEND MAXLEN RDREC RLOOP EXIT INPUT MAXLENZ WRREC	LDA STA STA STA STA STA STA STA RESW RESW LTORG =C'EO RESB EQU CSECT EXTREF CLEAR CLEAR CLEAR CLEAR CLEAR CLEAR TD JEQ STCH STCH TTD JEQ COMPR JEQ STCH STCH TTXR JLT +STCH TXR SUB BYTE WORD CSECT EXTREF CLEAR LDT LST LST LST LST LST LST LST LST LST LS	110010 010000 110010 110010 110000 110000 110010 110010 110010 110010 110000 110000 110000 110001 110010 110010 110010 110010 110010 110010	BUFFER #3 LENGTH @RETADR 1 1 4096 * BUFEND-BUFFI BUFFER, LENGY A S #MAXLEN2 INPUT RLOOP INPUT A,S EXIT BUFFER, X T RLOOP LENGTH X'F1' BUFEND-BUFFI LENGTH, BUFFI LENGTH, BUFFI LENGTH =X'05' WLOOP BUFFER, X	ER TH, BUFEND		010003 0F2006 3E2000 B410 B4400 B4400 B4400 B4400 B4400 5100029 E3201B 332FFA DB2015 A004 332009 5790003A B850 3B2FE9 13100029 4F0000 F1 B410 77100029 E32012 332FFA 5390003A	
106 107 108 109 110 111 112 113 114 115 116 117 122 123 124 125 126 127 128 129 130 131 134 135 136 137 138 139 139 130 131 134 135 136 137 138 139 139 130 130 130 130 130 130 130 130 130 130	14 15 16 17 18 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50	0023 01 01 01 01 01 01 01 01 01 01 01 01 01	#### Error: 023 024 029 026 029 026 03A 103 103 100 000 000 0000 0001 0010 0113 0115 0118 0116 0118 0117 0118 0118 0119 0119 0100 0100 0100 0100	RETADR LENGTH * BUFFER BUFFER BUFEND MAXLEN RDREC RLOOP EXIT INPUT MAXLENZ WRREC	LDA STA STA STA STA STA STA STA STA STA ST	110010 010000 110010 100010 110000 110000 110000 110010 110010 110010 110010 110000 110000 110000	BUFFER #3 LENGTH @RETADR 1 1 4096 * BUFFEND-BUFFI BUFFER, LENGY RLOOP INPUT A,S EXIT BUFFER, X T RLOOP LENGTH X'F1' BUFEND-BUFFI X LENGTH, BUFFI X LENGTH, BU	ER TH, BUFEND		010003 0F2006 3E2000 B410 B4400 B440 75100029 E32018 332FFA DB2015 A004 332009 5790003A B850 3B2FE9 13100029 4F0000 F1 B410 77100029 E32012 332FFA 5390003A DF2008	
106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 131 132 131 132 133 134 135 137 138 139 139 139 139 139 139 139 139 139 139	14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 30 31 33 34 35 36 37 38 39 40 41 42 43 44 44 45 46 47 48 49 49 49 49 49 49 49 49 49 49 49 49 49	0023 0000 000 000 000 000 000 000 000 00	#### Error: 0220 #### Error: 0226 0229 02C 03A 103 103 100 000 000 000 000	RETADR LENGTH * BUFFER BUFFER BUFEND MAXLEN RDREC RLOOP EXIT INPUT MAXLENZ WRREC	LDA STA STA STA STA STA STA STA RESW RESW LTORG =C'EO RESB EQU CSECT EXTREF CLEAR CLEAR CLEAR CLEAR CLEAR +LDT JEQ STCH TIXR JLT +STX RSUB BYTE WORD CSECT EXTREF CLEAR LEAR LEAR LEAR LEAR LEAR LEAR LEAR	110010 010000 110010 110010 110000 110000 110010 110010 110010 110010 110000 110000 110000 110001 110010 110010 110010 110010 110010 110010	BUFFER #3 LENGTH @RETADR 1 1 4096 * BUFEND-BUFFI BUFFER, LENGY A S #MAXLEN2 INPUT RLOOP INPUT A,S EXIT BUFFER, X T RLOOP LENGTH X'F1' BUFEND-BUFFI LENGTH, BUFFI LENGTH, BUFFI LENGTH =X'05' WLOOP BUFFER, X	ER TH, BUFEND		010003 0F2006 3E2000 B410 B4400 B4400 B4400 B4400 B4400 5100029 E3201B 332FFA DB2015 A004 332009 5790003A B850 3B2FE9 13100029 4F0000 F1 B410 77100029 E32012 332FFA 5390003A	
106 107 108 109 110 111 112 113 114 115 116 117 122 123 124 125 126 127 128 129 131 132 133 134 135 136 137 138 139 140 141 141 141 141 141 141 141 141 141	14 15 16 17 18 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 51 51 51 51 51 51 51 51 51 51 51 51	0023 01 01 01 01 01 01 01 01 01 01 01 01 01	#### Error: 0229 0220 024 025 029 026 029 026 03A 103 103 100 000 000 000 000 000 001 013 015 018 000 000 000 000 000 000 000 000 000	RETADR LENGTH * BUFFER BUFFER BUFEND MAXLEN RDREC RLOOP EXIT INPUT MAXLENZ WRREC	LDA STA STA STA STA STA STA STA STA STA ST	110010 010000 110010 110010 110010 110000 110010 110010 110010 110010 110010 110010 110010 110010 110010 110010 110010 110010 110010 110010 110010 110010	BUFFER #3 LENGTH @RETADR 1 1 4096 * BUFFEND-BUFFI BUFFER, LENGT RLOOP INPUT A,S EXIT BUFFER, X RLOOP LENGTH X'F1' BUFEND-BUFFI LENGTH, BUFFER, X LENGTH =X'05' WLOOP BUFFER, X =X'05' T WLOOP	ER TH, BUFEND		010003 0F2006 3E2000 B410 B4400 B4404 75100029 E32018 332FFA DB2015 A004 332009 5790003A B850 3B2FE9 13100029 4F0000 F1 B410 77100029 E32012 332FFA 5390003A DF2008 B850 3B2FEE 4F0000	
106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 131 132 133 134 135 136 137 138 139 139 139 139 139 139 139 139 139 139	14 15 16 17 18 120 21 223 24 25 26 27 28 30 31 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 50 50 50 50 50 50 50 50 50 50 50 50	0023 01 01 01 01 01 01 01 01 01 01 01 01 01	#### Error: 022 022 022 022 022 022 022 0	RETADR LENGTH * BUFFER BUFFER BUFEND MAXLEN RDREC RLOOP EXIT INPUT MAXLENZ WRREC	LDA STA STA STA STA STA STA STA STA RESW RESW LTORG =C'EO RESB EQU CSECT EXTREF CLEAR CLEAR CLEAR CLEAR CLEAR TD JEQ STA STCH TIXR JLT +STX RSUB BYTE WORD CSECT EXTREF CLEAR LEAR LDT LT	110010 010000 110010 100010 110000 110000 110000 110010 110010 110010 110010 110010 110010 110010 110010 110010 110010 110010 110010	BUFFER #3 LENGTH @RETADR 1 1 4096 * BUFEND-BUFFI BUFFER, LENG X A S #MAXLEN2 INPUT RLOOP INPUT A, S EXIT BUFFER, X T RLOOP LENGTH X'F1' BUFEND-BUFFI LENGTH, BUFFER, X LENGTH =X'05' WLOOP BUFFER, X =X'05' T	ER TH, BUFEND		010003 0F2006 3E2000 B410 B4400 B4400 B4400 B4400 5100029 E32018 332FFA DB2015 A004 332009 5790003A B850 3B2FE9 13100029 4F0000 F1 B410 77100029 E32012 332FFA 5390003A DF2008 B850 3B2FEE	

File: /home/emil/Desktop/SICXEAssembler/OBJFILE.o

```
HCOPY ^000000^00103A
1
    T000000^1D^1720234B10103A03A01F2980003320074B10002C3F2FEC0320120F201D
   T00001D^0C^0100030100030F20063E2000
    M000004^05+RDREC
    M000011^05+WRREC
 6
   E000000
8
9
   HRDREC ^000000^00002C
10
    DBUFFER00003ABUFEND00103ALENGTH000029
11
12
    RRDREC WRREC
    T000000^1E^B410B400B44075100029E3201B332FFADB2015A0043320095790003AB850
13
   T00001E^0E^3B2FE93B2FE9131000294F0000F1
14
    M000007^05
15
   M000019^05+BUFFER
16
    M000022^05+LENGTH
17
    M000029^06+BUFEND
18
    M000029^06-BUFFER
19
20
    E000000
21
22
23
   HWRREC ^000000^00001B
25
    RBUFFERLENGTHBUFEND
    T000000^1B^B41077100029E32012332FFA5390003ADF2008B8503B2FEE4F0000
26
    M000003^05+LENGTH
27
28
   M00000D^05+BUFFER
29 E000000
```