SIC/XE Assembler Report

Author: [Mohamed Mahmoud Elbagoury, Habibah Mohamed Afifi]
Date: [13/05/2025] **To**: [Eng.Eman Maher]

1. Overview of Key Functions

1.1 getInput(string l, string *a, string *b, string *c)

- **Purpose**: Parses a line of assembly code into **label**, **opcode**, and **operand**.
- **Example**:

```
"FIRST STL RETADR" → Label="FIRST", Opcode="STL", Operand="RETADR"
```

1.2 wrdsize(int i)

- **Purpose**: Computes byte size for `WORD` directives.
- **Logic**:

```
"WORD 1234" → 2 bytes (4 hex digits / 2 = 2 bytes).
```

1.3 bytsize(int i)

- **Purpose**: Computes byte size for `BYTE` directives.
- **Logic**:

```
"BYTE C'EOF'" → 3 bytes (ASCII chars 'E', 'O', 'F').
```

1.4 hextoint(string hexstring)

- **Purpose**: Converts hex strings to integers.

```
"1A" → 26 (decimal)
```

1.5 inttohex(int x, int b)

- **Purpose**: Converts integers to fixed-width hex strings.

```
inttohex(26, 4) → "001A"
```

1.6 `bintohex(bool a, bool b, bool c, bool d)`

- **Purpose**: Encodes 4 flags into 1 hex byte.

```
bintohex(1,0,1,0) → "A" (8 + 2 = 0xA)
```

1.7 `readdr(string res)`

- **Purpose**: Computes PC-relative or base-relative addresses.
- **Example**:

```
PC=1000, target=1020 → Displacement = "014" (hex).
```

1.8 `format2(int i)`

- **Purpose**: Generates object code for **Format 2** (register) instructions.

```
"CLEAR X" → "B4" (opcode) + "1" (X reg) → "B41"
```

1.9 `format3(int i)`

- **Purpose**: Handles **Format 3** (standard) instructions.
- **Features**:
  - Indirect (@), Immediate (#), Indexed (X).

```
"LDA BUFFER,X" → "032026"
```

1.10 `format4(int bb)`

- **Purpose**: Generates **Format 4** (extended) instructions.

```
"+JSUB RDREC" → "4B101036" (4-byte).
```

1.11 `ObjectFile()`

- **Purpose**: Writes object code in SIC/XE format.
- **Output**:

```
H^COPY  ^000000^001077
T^000000^10^17202D^69202D^4B101036...
```

1.12 `printsymtab()`

- **Purpose**: Generates a sorted symbol table.
- **Output**:

```
+------------+------------+
| Symbol     | Address    |
+------------+------------+
| CLOOP      | 0006       |
| ...        | ...        |
+------------+------------+
```

2. Two-Pass Assembler Workflow

Pass	Task
1	Builds `SYMTAB` and tracks `LOCCTR`.
2	Generates object code using `OPTAB`/`SYMTAB`.

3. Output Files

File	Description
`SICXE_SYMBOL_TABLE.txt`	Symbol addresses.
`SICXE_Output_LOCATION.txt`	Instructions with object code.
`SIC_XE_Out.txt`	Final object code.

4. Conclusion

This assembler:
✅ Supports all SIC/XE addressing modes.

✅ Generates object code, symbol tables, and listings.
✅ Follows the classic two-pass design.

End of Report
