Trivial Language for Z80

Eric Neblock

Last updated: August 8, 2016

1 Introduction

This document will give a brief overview of how the *Trivial Language for Z80* works and how to actually program in it. program in it.

2 Reserve Words and Keywords

There are several reserve words within the language. They are as follows:

2.1 Reserve Words

The folling list is to demostrate which words can not be used in the language other than their intended purpose.

1. main()
2. send_int(ADDRESS, int)
3. send_char(ADDRESS,char)
4. send_int_array(ADDRESS, int[], size)
5. send_char_array(ADDRESS, char[], size)
6. read_int(ADDRESS, int)
7. read_char(ADDRESS, char)
8. read_int_array(ADDRESS, int[], size)

9. read_char_array(ADDRESS, char[], size)
10. for
11. while
12. return
13. function

2.2 Data Types

The following are data types that exist in the language. These can not be overridden or expanded.

1. int
2. char
3. float
4. array → char[constant]

Variables must start with a character (case doesn't matter) and **can not** start with an _.

Identifiers are case insensitive within their scope. So, if you have a variable called *var* in main, then it can be referred to *var, Var, VAR, vAr, and so on.*

Each function creates a new scope, leaving you the option to have variables called the same thing and have different meaning in functions (*main() can have "char x" and fun1() can have "int x"*)

2.3 Keywords

The following are data types that deal with interacting with the data bus.

1. ADDRESS
2. WRITE
3. READ

2.4 Comments

Our language has only one way to enter comments and that is in C-style (/* *comment* */)

3 Syntax

In this language, we have certain rules on how to have things up and running.

For example, there can only be one statement per line – with some exceptions. Blank lines are ignored. The assembly will put comments at the start of a block, so:
`int x = 0 /* set a variable */` will turn into:

```
x defb 0 ; set a variable
```

4 Sample Programs

```
 main()
{
char type[11]
type = ‘‘hello world"
send_char_array(0xFF00, type, size)
}

function fun(int x)
{
x = x + 1
return x
}
main()
{
int x
x = 11
x = fun(x)
send_int(0x00FF, x)
}
```