Four-Bit ALU Trainer Simulator User's manual

Author: Rodney Moffitt

ABSTRACT

This document describes how to use to program '4 Bit ALU Trainer Simulator', which provides software simulation for the '4 Bit ALU Trainer'.

TABLE OF CONTENTS

ABSTRACT 2
1 OVERVIEW 4
2 SYSTEM REQUIREMENTS 5
3 INSTALLING THE PROGRAM 6
4 RUNNING THE PROGRAM 7
5 USER INPUT/OUTPUT 10
6 CONTROL MEMORY 12
6.1 Loading 12
6.2 Over-Riding 12
7 EXECUTING AN INSTRUCTION 14
REFERENCES 15

1 OVERVIEW

The '4 Bit ALU Trainer Simulator' is a Windows 3.1 application that allows simulation of the hardware version of the '4 Bit ALU Trainer'.

All features of the trainer are provided for in the simulator including the additional ability of altering the control program memory. It is the purpose of the trainer to allow students the ability to execute instructions without the need of the hardware version.

2 SYSTEM REQUIREMENTS

In order to run the simulator, a PC running Windows 3.1 is required.

A number of support files, including dynamic link libraries are required to run the program, these are:

1. LOGIC.EXE
2. CONTRMEM.TXT
3. \WINDOWS\SYSTEM\CMDIALOG.VBX
4. \WINDOWS\SYSTEM\SPIN.VBX
5. \WINDOWS\SYSTEM\THREED.VBX
6. \WINDOWS\SYSTEM\VBRUN300.DLL

'LOGIC.EXE' is the simulator program. 'CONTRMEM.TXT' is a text representation of the control memory required by the program. The .VBX and .DLL files are support files which must be either in the \WINDOWS\SYSTEM\ directory or in the same directory as 'LOGIC.EXE'.

A distribution disk which has all of these files, plus a program to install them in their appropriate directories is available. It is described in the following section.

3 INSTALLING THE PROGRAM

A distribution disk which includes all of the required files and a set-up program is available. The following steps will install the simulator using this disk:

1. Place the distribution disk into a floppy drive.
2. From the FILE menu in the Windows Program Manager, choose RUN.
3. Select BROWSE and then choose the file 'SETUP.EXE' from the floppy drive.
4. Select OK from the BROWSE dialogue box.
5. Select OK from the RUN dialogue box.

The program SETUP.EXE will install all files including creating a program group with an icon for LOGIC.EXE and a directory on the local hard drive. To start the program double-click on the icon.

4 RUNNING THE PROGRAM

Upon starting the simulator a window is displayed with the title bar caption '4 Bit ALU Trainer Simulator'. The window contains block elements that are used in the hardware trainer. The following figure is a screen capture of the simulator running with a screen size of 1024x768 (other resolutions will display slightly different).

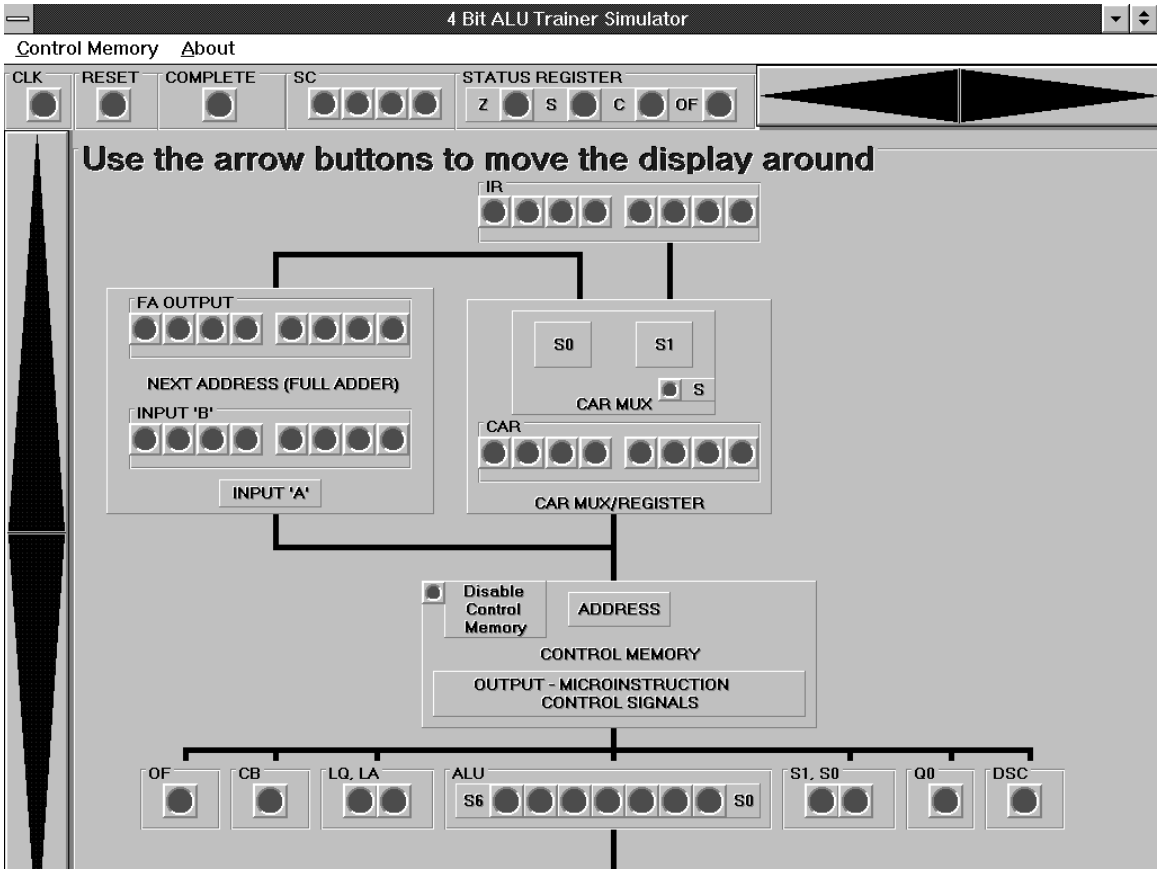

Figure 1: Simulator Window

All the elements of the hardware version of the trainer have been implemented in the software simulator.

There are four main parts to the simulator:

1. Menu bar.

The menu mar has the menu item 'Control Memory'. Clicking on it displays a sub-menu with the options 'Display' and 'Load From File' as choices. Clicking on 'Display' shows another form which is seen in Figure 2 below. The 'About' menu item displays information about the program.

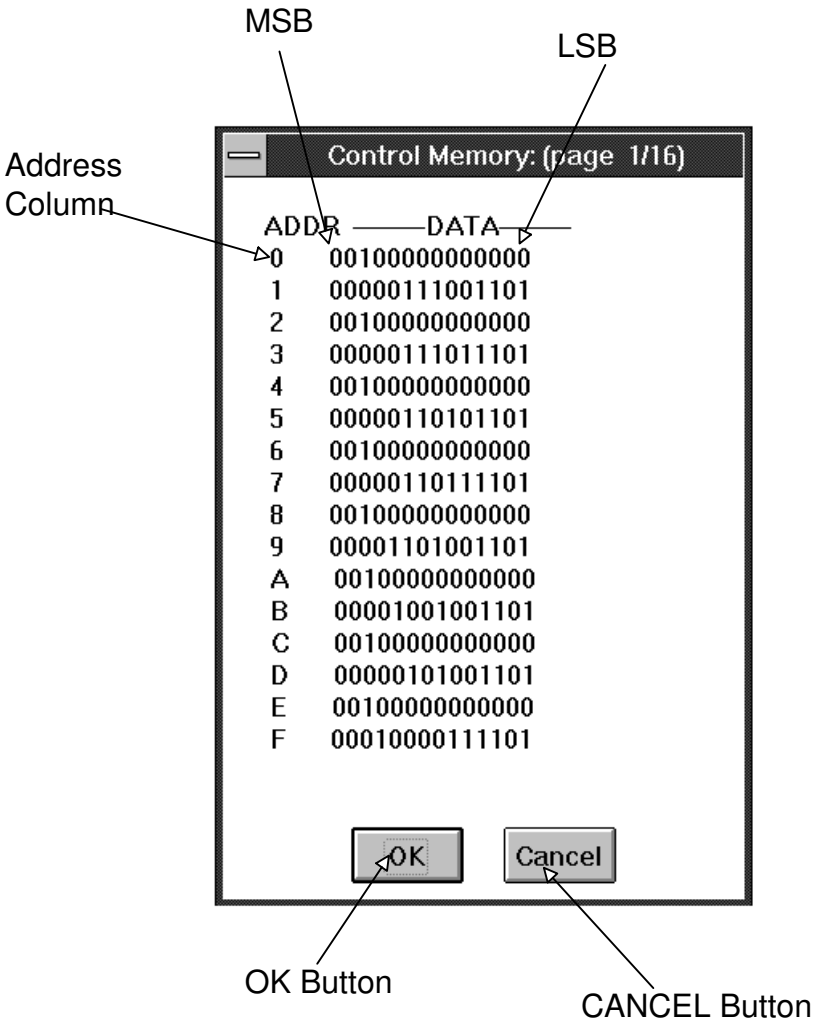

Figure 2: Control Memory Display Window

This form displays the control memory, sixteen micro-instructions at a time. The address column is displayed in hexadecimal. The MSB to the left corresponds to the 'OF' bit and the LSB to the right is the 'DSC' bit.

Clicking on 'OK' displays another sixteen micro-instructions, 'CANCEL' removes the form and discontinues the display of the control memory.

By clicking on 'Load From File' displays a dialogue box requesting a file name which the simulator will load as the control memory (see the section 'Control Memory' for more details).

2. Navigation/Arrow Buttons

Due to the large size of the simulator screen it is not possible to display all of the features at once. It is possible to change what part of the display is shown by the mouse or the keyboard.

The vertical and horizontal arrow buttons allow the display frame to be moved around the screen. There are also key commands:

- HOME causes the screen to show the top of the display frame.
- END causes the screen to show the bottom of the display frame.

Having the NUMERIC LOCK ON also allows the 4,8,6,2 numeric keypad arrow keys to move the display around the same way the arrow buttons would. If the arrow buttons do not display (due to a compatibility problem with the software, or not enough room due to low resolution) then it will be required to solely use the keyboard to navigate the display.

3. Status display.

The CLK, RESET, FINISH, SC and STATUS REGISTER signals are displayed in the upper-left corner of the screen. These elements are not part of the display frame and therefore never move from their position. The user can then know the status of these important signals no matter what part of the display is currently being shown.

4. Display frame.

The display frame is the 'container' of all the elements of the trainer. The display frame can be moved (when the navigation buttons/keys are pressed) since it is larger than the screen.

5 USER INPUT/OUTPUT

All user input that would have occurred with switches and LEDs in hardware are done by graphic input/output LEDs in the simulator.

The following figure shows some signals grouped together in a 'frame', with associated LEDs:

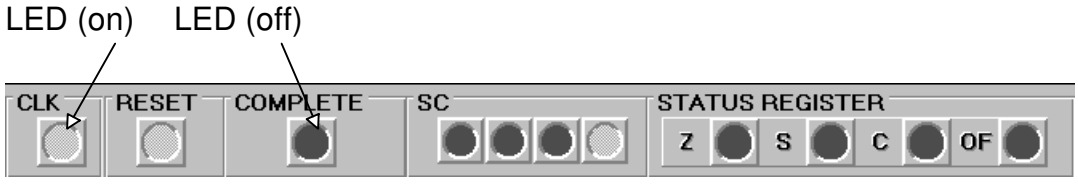

Figure 3: Input/Output Using LEDs

The signals are grouped together in frames to make the user interface easier to use. When a signal has a value 'TRUE' then the associated LED is green, and when 'FALSE' it is red. The LEDs that are associated with signals that the user can alter are clicked on to toggle the value. Not all LEDs are meant to be input signals so clicking on output signal LEDs will have no effect.

The user can click on (and thus change the value of the associated signal) the following LEDs:

1. CLK
2. RESET
3. IR
4. ALPHA
5. BETA
6. GAMMA
7. Micro-instruction Control Signals

A special case arises for the micro-instruction control signals (ALU, CB, etc.) which both display values and allow the user to change them. In the hardware version of the trainer the control memory can be overridden to so that the output of the EPROM has no effect on the transfer logic and registers. This is also provided for in the simulator.

To change the micro-instruction bits the 'Disable' LED must first be set (which is inside the control memory element). This allows the user to click on the control signal LEDs and inhibit further changes to the control signals by the value CAR. By clicking on the LED again, causing it to go to 'FALSE', will inhibit any changes to the control signals by the user and enable further changes to these signals by the value of CAR.

An additional input has been employed to allow CLK to be toggled more efficiently. In addition to the mouse clicks on the LED 'CLK', the user can also press the space bar to toggle the clock signal.

6 CONTROL MEMORY

6.1 LOADING

The control memory is storage of all the individual micro-instructions that make up the instructions. The user can alter the control memory by changing the file 'CONTRMEM.TXT' which is first loaded when the program is started.

The user can change the file and force the simulator to re-load the control memory by clicking on the menu 'Control Memory' then the sub-menu 'Load From File'. A dialogue box prompts the user for a file name.

The format of the file is a string of fifteen ones/zeros which define the micro-instruction control signals, the 17th character is either a 'D', 'M' or another character (or blank). Anything past the 17th character is ignored (so comments can be placed there). The 16th character should also be blank. Only one micro-instruction per line is allowed. An example is displayed here:

```
111110000000000 blah1
000010110101001 blah2
010110101010100 D
010110101010100
101010010011011 M
```

If this were the beginning of the file then the line with comment '`blah1`' would be loaded into address 0000H, and the next line '`blah2`' would be loaded into address 0001H, and so on.

The 17th character is important if it contains D or M since these signify that the micro-instruction on that line is either the address of the divide or multiply instructions, respectively. These addresses are important as the arithmetic processor in the trainer has special conditions for these instructions. For example the line after the address 0001H line will be considered the start of the divide instruction, and address 0005H will be considered the first micro-instruction of the multiply instruction.

If there is a problem loading the control memory from a file the simulator will insert zeros in the control memory after the point the error was encountered.

6.2 OVER-RIDING

There may be times when the control memory must be disabled so that the control signals can be driven by the user only. This is described in the section 'User Input'. If the 'Disabled' LED is not set then subsequent rising edges on CLK (clicking from 0 to a 1)

will cause the CAR to be loaded with a new value and a new set of micro-instruction control signals to be loaded.

7 EXECUTING AN INSTRUCTION

The simulator is used in the same way the hardware version is used with CLK providing synchronization for the entire circuit.

To initiate an instruction the steps outlined in the following diagram are executed:

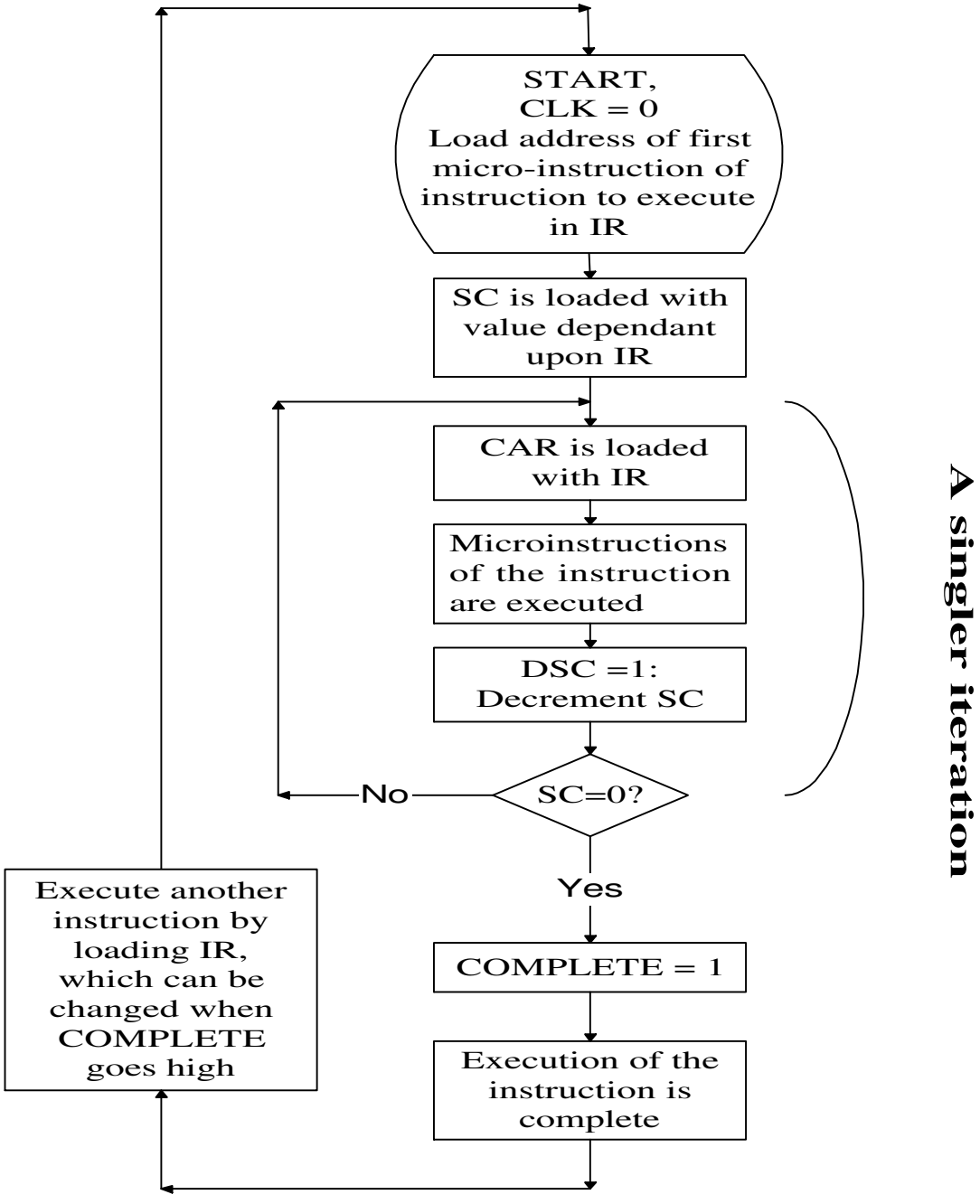

Figure 6: Sequence For Executing An Instruction

REFERENCES

1 Moffitt, Rod S., "Four-Bit ALU Trainer Detailed Design Document", 1995