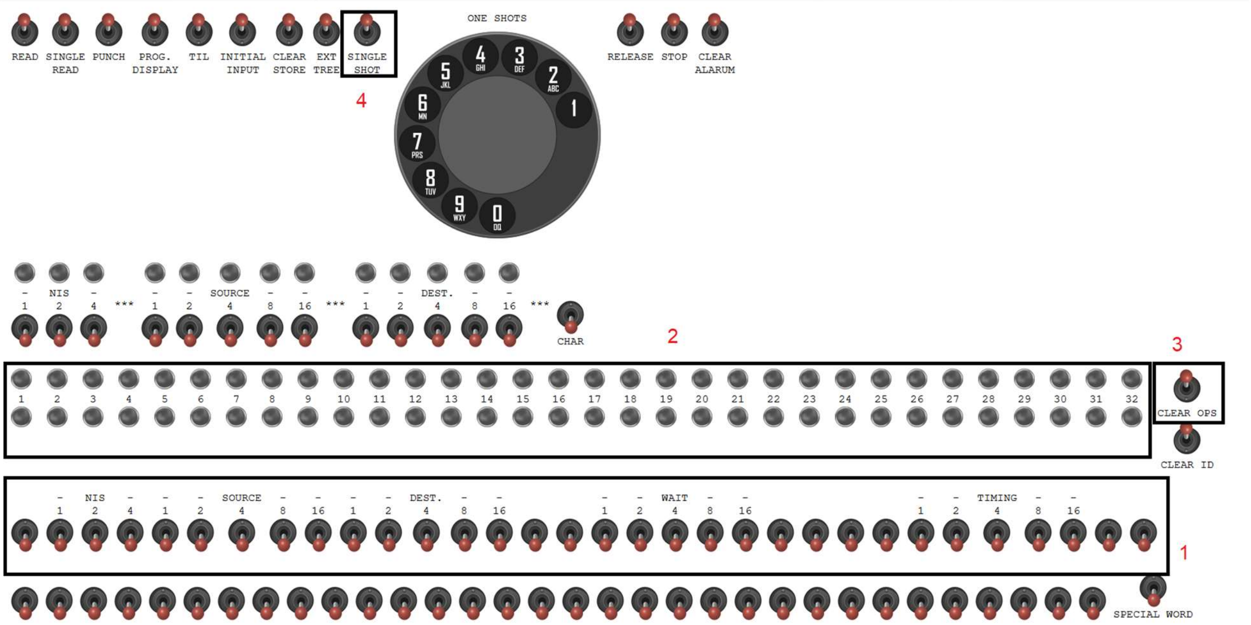
DEUCE Emulator User Guide

Welcome to the DEUCE Emulator user guide! The DEUCE was one of the world's first commercially available computers and this emulator aims to recreate the system within a web application. The purpose of this project is to continue the legacy of the DEUCE and allow people to better understand how early computers functioned.

To get started, let's go over the key emulator functions you will be using in this tutorial. Go to https://gerardward3.github.io/level4-project/ and the emulator should load in your web browser. For the purpose of this tutorial, here are the key parts of the emulator you will need to know about:



1. **The Input Dynamiciser**
   This row of switches allows input to be read into the DEUCE one instruction at a time. From here, you can provide the DEUCE with a 32-bit word, which can be an instruction or a reverse binary number. For example, clicking on the first three left-most switches on this row sets the Input Dynamiciser to 7. This is because these switches correspond to 1, 2 and 4 in reverse binary, so 1 + 2 + 4 = 7.
   Instructions are broken down into the following parts as labelled on the console: Next Instruction Source, Source, Destination, Wait and Timing. These individual parts are also in reverse binary format. Essentially, the DEUCE carries out operations by performing a series of transfer instructions using these special parts of the instruction. In this guide, the parts of the instruction being focused on the most will be the Source and Destination parts. This will be explained in more detail later in this guide.

2. **The Output Staticiser + Input Dynamiciser lights**
   These lights are used to display output on the DEUCE. The top row in this section is the Output Staticiser, which shows output when an instruction is executed. The second row displays the status of the Input Dynamiciser. When a switch on the Input Dynamiciser is clicked, its corresponding light is switched on.

3. **Clear Ops Switch**
   This clears the Output Staticiser, setting all lights on this row to off.

4. **Single Shot Switch**
   This switch is used to execute a single instruction. When an instruction is given through the Input Dynamiciser, clicking the Single Shot switch processes this instruction.

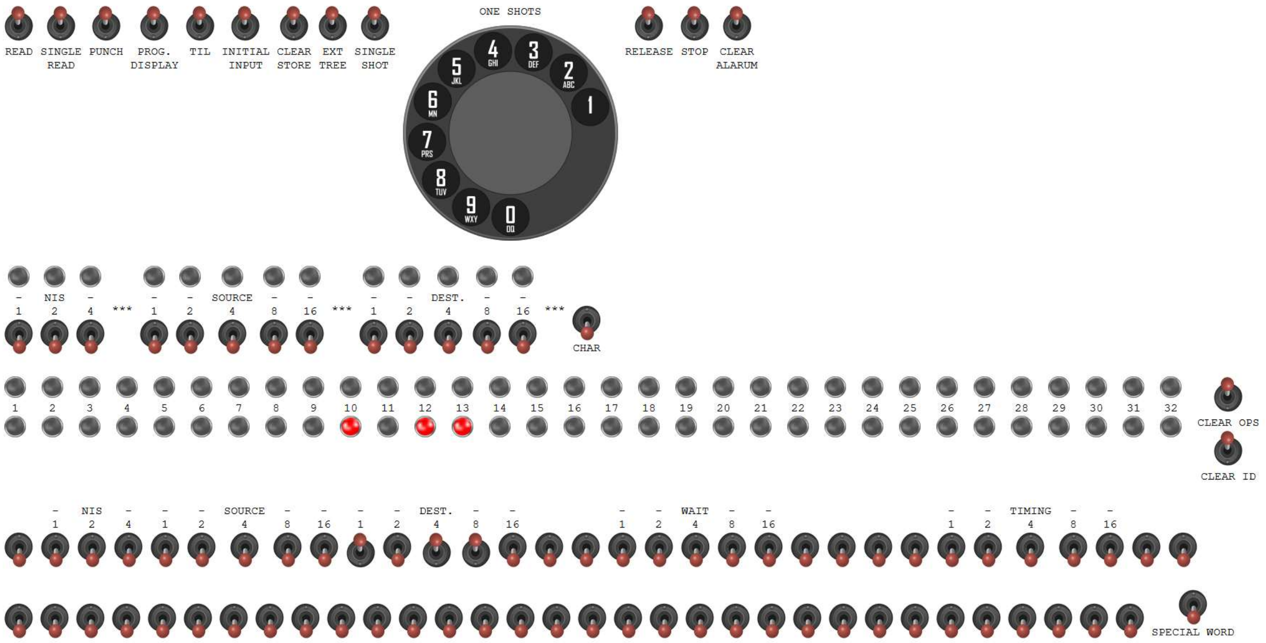
   If at any point the emulator fails to work as expected, please click refresh to start this tutorial over again.

Writing your first DEUCE program: Adding two numbers

The following program will demonstrate how to give the DEUCE data via the Input Dynamiciser and perform binary addition using the console.

1. **Instruction 0 – 13**

   Firstly, set the Destination field of the Input Dynamiciser to 13 (Dest. 1 + 4 + 8) so it resembles the following:



   Destination 13 is a Temporary Store, which means it can hold one 32-bit word.

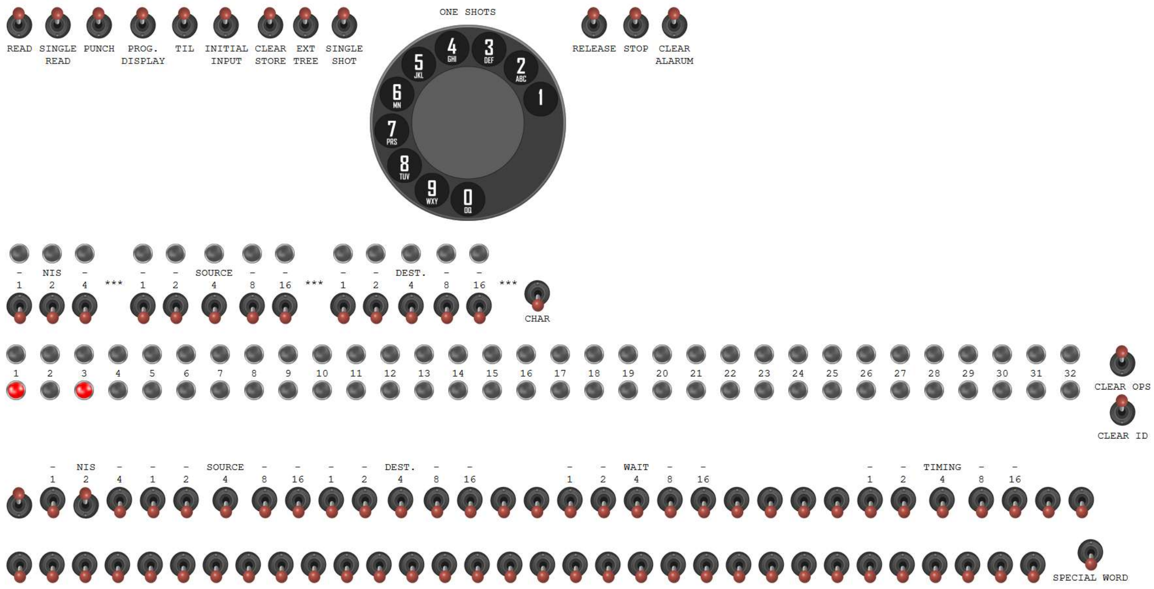
   The Source of this instruction is left to 0. This is because Source 0 is a special destination that allows a number to be read in from the Input Dynamiciser after this instruction has been executed. Now, your instruction is ready to be read in by the computer

2. **Click the Single Shot switch**

   Now that the instruction has been set up correctly on the Input Dynamiciser, click the **Single Shot** switch to execute your instruction.

3. **Send decimal number 5 to Temporary Store 13**

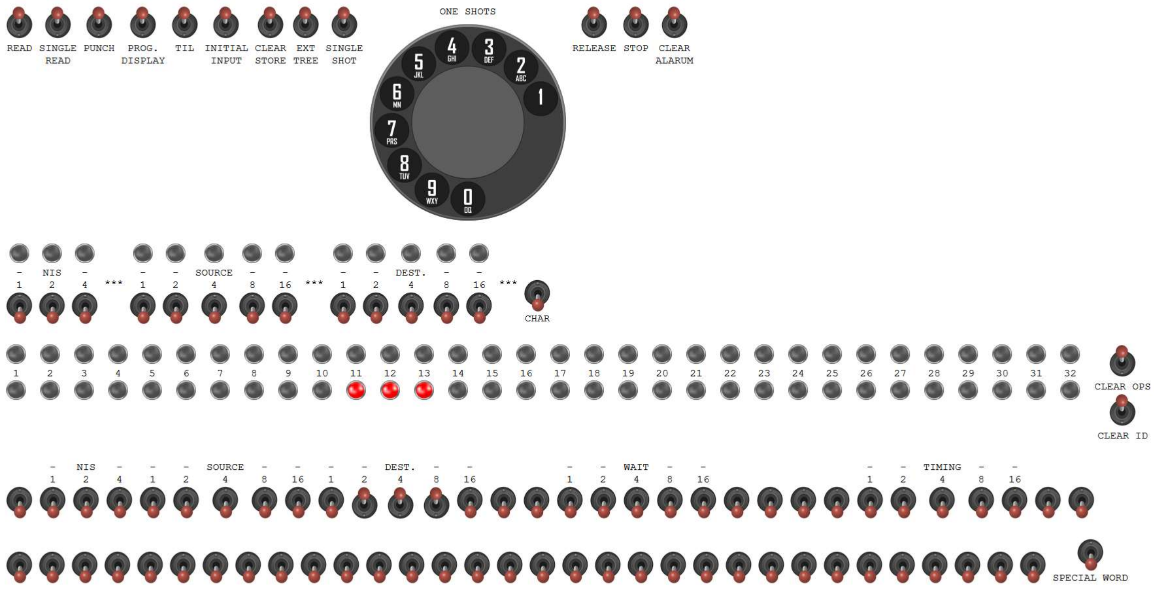
   Instruction 0 – 13 has been executed, so now a 32-bit number can be sent to Temporary Store 13. This number can be entered using the Input Dynamiciser. To send the number 5 to TS13, make sure the switches that were on before are turned off and turn on the following switches:



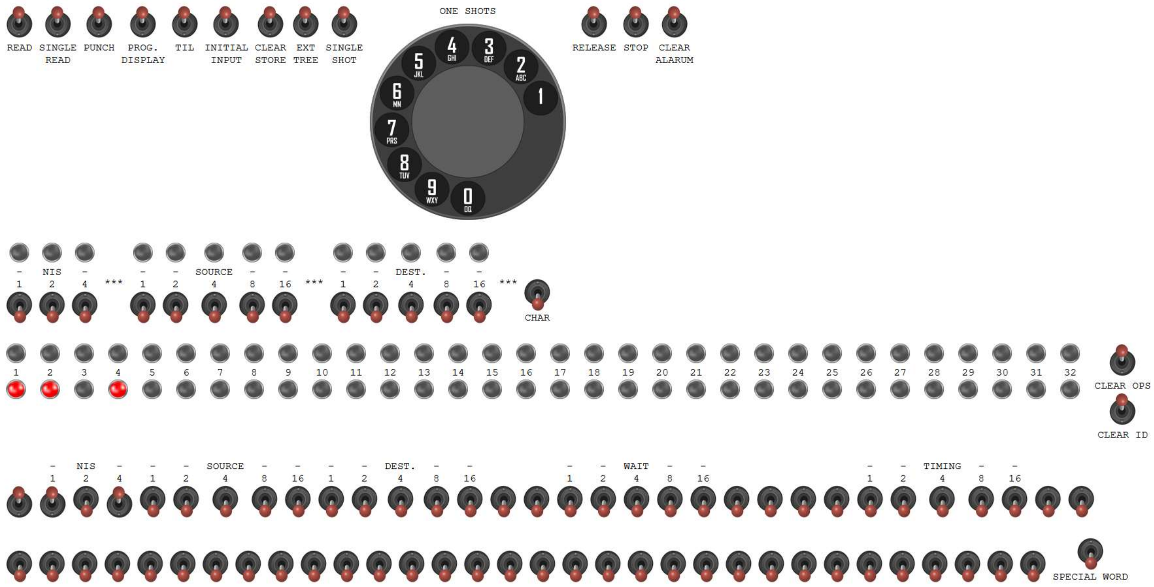
   This is the number 5 in reverse binary. Like before, when you have entered this on the Input Dynamiciser, click **Single Shot** to put 5 in TS13.

4. **Send 11 to TS14.**

   Now, Temporary Store 13 holds a value of 5. We are going to add 11 to it to make a total of 16 and display this on the Output Staticiser. First, we will repeat the previous steps but with slight differences. Set the Input Dynamiciser so it holds instruction 0 – 14, like so:



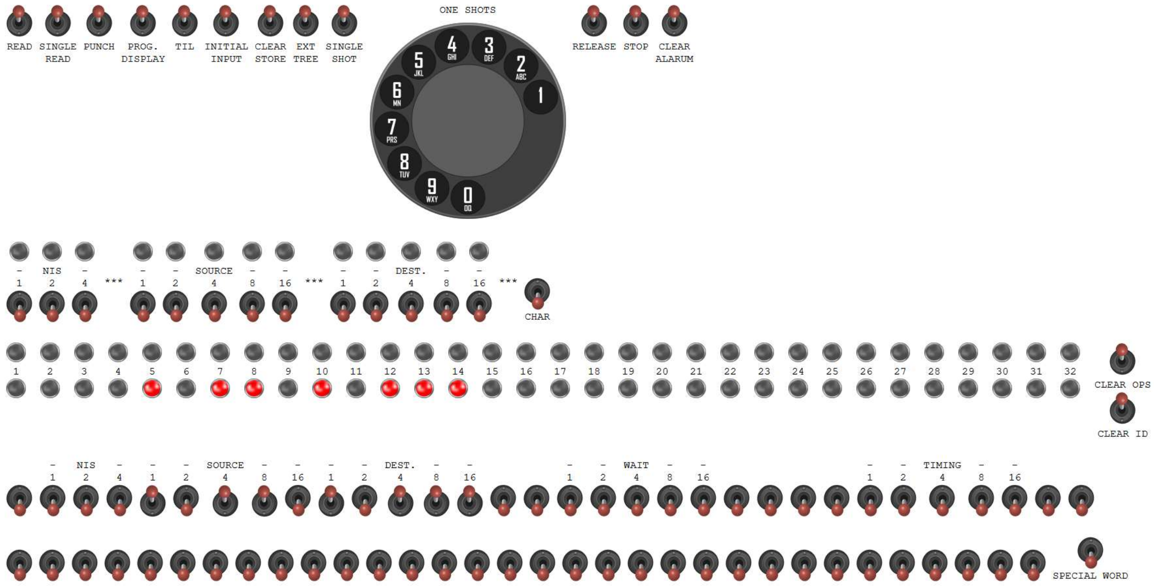
Then, click the **Single Shot** switch. Now, enter reverse binary number 1101 (decimal number 11) on the Input Dynamiciser:



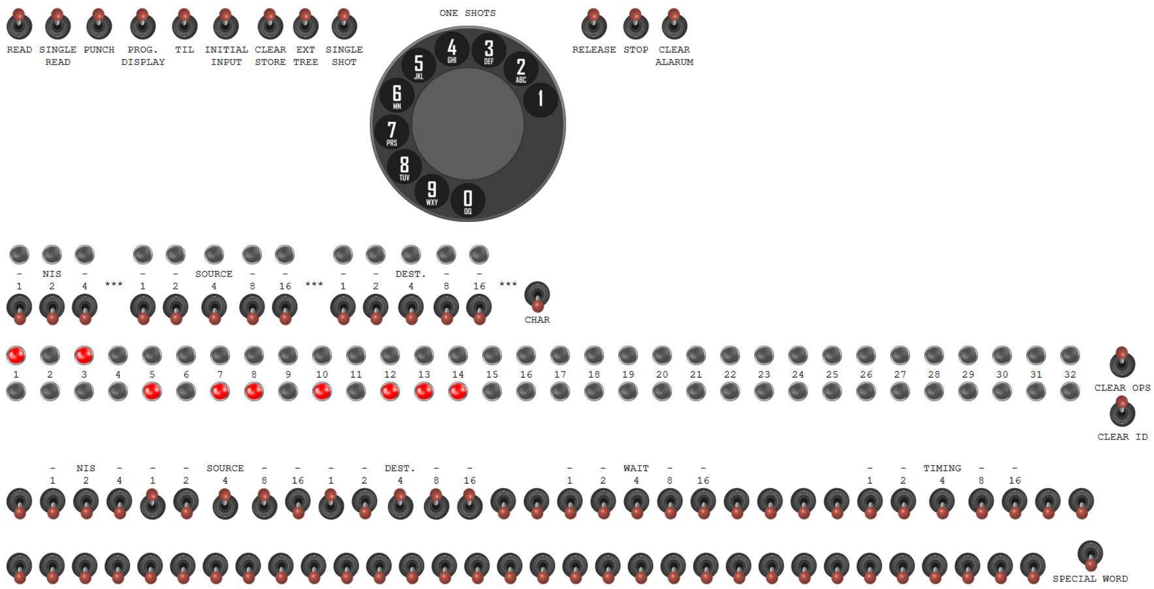
Click **Single Shot** again to execute this instruction. Now, Temporary Store 13 should hold 5 and Temporary Store 14 should hold 11.

5. Check output of Temporary Stores

Like Source 0, Destination 29 is another special address that displays the contents of a memory address on the Output Staticiser. To check the contents of Temporary Store 13, enter the instruction 13-29 on the Input Dynamiciser:



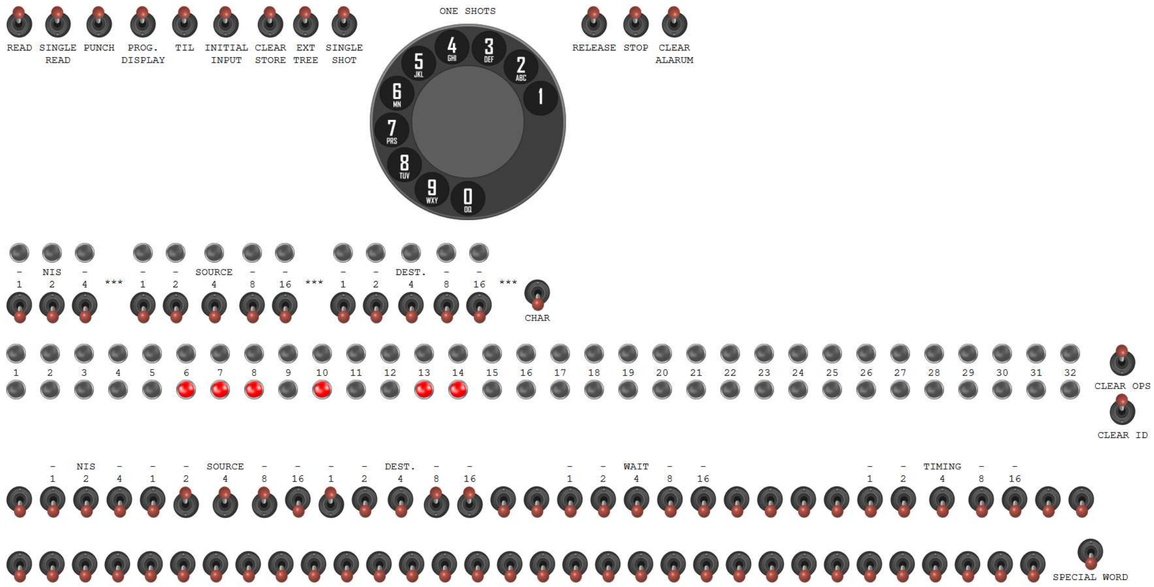
This sets the Source value to 13 (1 + 4 + 8) and the Destination value to 29 (1 + 4 + 8 + 16). By clicking **Single Shot**, the following should appear on screen:



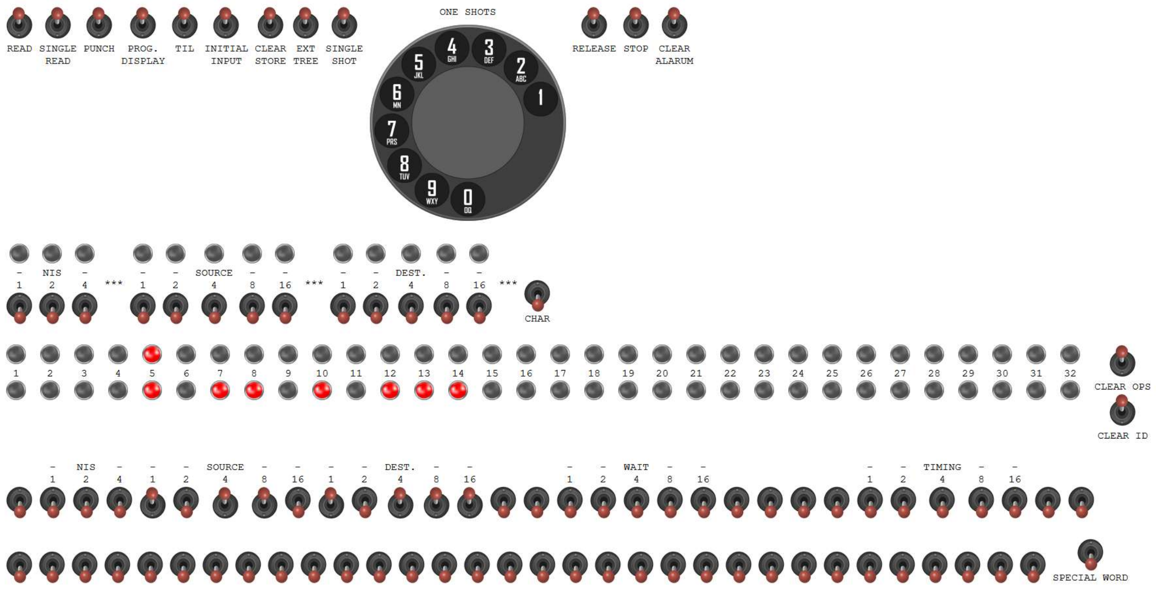
This displays the value 5 on the Output Staticiser, which is the value held in TS13. Likewise, the contents of TS14 can be viewed by using the same method, only changing the Source value to 14 (2 + 4 + 8). Click the Clear Ops switch to reset the lights on the Output Staticiser to 0.

6. Add the contents of TS13 and TS14

We will now add the contents of the two memory locations we have loaded with values. To perform addition, we can use Destination 25 as another special memory location. This location adds the contents of the Source location given to Temporary Store 13. Therefore, we can give our Source location as 14 and add the contents of TS14 to TS13. To do this, enter Instruction 14 – 25 (as pictured below) on the Input Dynamiciser and click **Single Shot**:



Now, TS13 should hold a value of 16. To check this, enter instruction 13 – 29 on the Input Dynamiciser, click **Single Shot**, and the following should appear on your screen:



The Output Staticiser now shows a value of 16. Using these instructions, you have now successfully created your first DEUCE program!

There are many more functions available in this emulator, but this is to give you an introduction to DEUCE programming. For example, you can perform subtraction by using Destination 26 instead of 25 in the previous steps. Hopefully, you now have a better understanding of how the DEUCE functioned as a computer. Thank you for using this emulator!

Special source and destination addresses

Sources

- Source 23 – Divides contents of TS14 by 2 and places in Destination address.
- Source 24 – Multiplies contents of TS14 by 2 and places in Destination address.
- Source 27 – Places 1 in Destination address.
- Source 28 – Places 2^{16} in Destination address.
- Source 29 – Places 2^{31} in Destination address.
- Source 30 – Places 0 in Destination address.
- Source 31 – Places -1 in Destination address.

Destinations

- Destination 25 – Adds contents of Source address to TS13.
- Destination 26 – Subtracts contents of Source address from TS13.
- Destination 27 – Checks if contents of Source address is negative or positive.
- Destination 28 – Checks if contents of Source address is zero or non-zero.
- Destination 29 – Displays contents of Source address on Output Staticiser lights.