

ECE1195 Lab 4 Multiplication Test: Joseph Schurer

Assembly Program

```
lui $2, 0x0000FFFF
addi $3, $0, 2
MULTU $2, $3
MFLO $5
MFHI $6
lui $1, 0x00001001
SW $5, 36($1)
SW $6, 52($1)
```

Tcl Script

```
# restart the simulation
Restart
```

```
add_wave {{/cpu_tb/U_1}}
```

```
# Forcing a program (instruction memory)
```

```
#lui $2, 0x0000FFFF
#addi $3, $0, 2
#MULTU $2, $3
#MFLO $5
#MFHI $6
#lui $1, 0x00001001
#SW $5, 36($1)
#SW $6, 52($1)
```

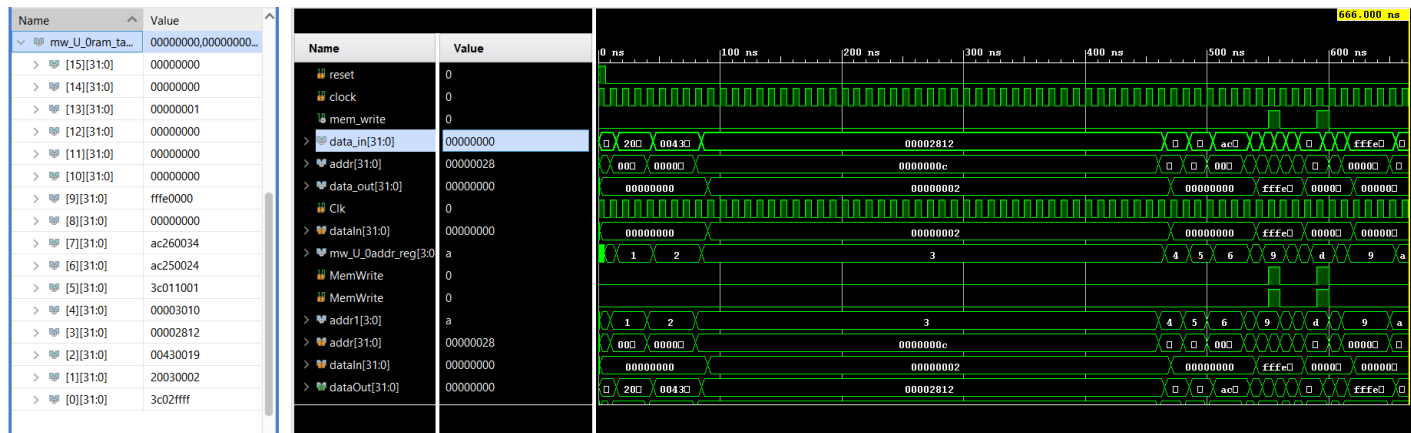
```
add_force {/cpu_tb/U_1/mw_U_0ram_table[0]} -radix hex {3C02FFFF}
add_force {/cpu_tb/U_1/mw_U_0ram_table[1]} -radix hex {20030002}
add_force {/cpu_tb/U_1/mw_U_0ram_table[2]} -radix hex {00430019}
add_force {/cpu_tb/U_1/mw_U_0ram_table[3]} -radix hex {00002812}
add_force {/cpu_tb/U_1/mw_U_0ram_table[4]} -radix hex {00003010}
add_force {/cpu_tb/U_1/mw_U_0ram_table[5]} -radix hex {3C011001}
add_force {/cpu_tb/U_1/mw_U_0ram_table[6]} -radix hex {AC250024}
add_force {/cpu_tb/U_1/mw_U_0ram_table[7]} -radix hex {AC260034}
```

```
#forcing a clock with 10 ns period
add_force clock 1 {0 5ns} -repeat_every 10ns
```

```
#give a reset signal
add_force Reset 0
run 1 ns
add_force Reset 1
run 5 ns
add_force Reset 0
```

```
run 660 ns
```

Waveform from Script



Discussion of Results

My multiplication instructions appear to be working correctly. I chose the inputs such that the result would be large enough to populate both the high and low registers. I was then able to store the results from the multiplication command after using MFLO and MFHI. The results can then be seen in 9 and 13 on the left side of the waveform.