CPE 233 Software Assignment 3

Decision Trees in Assembly

Report by:
Ethan Vosburg (evosburg@calpoly.edu)
Lawrence Nichols (lnicho06@calpoly.edu)

January 30, 2024

Table of Contents

1 Flow Charts

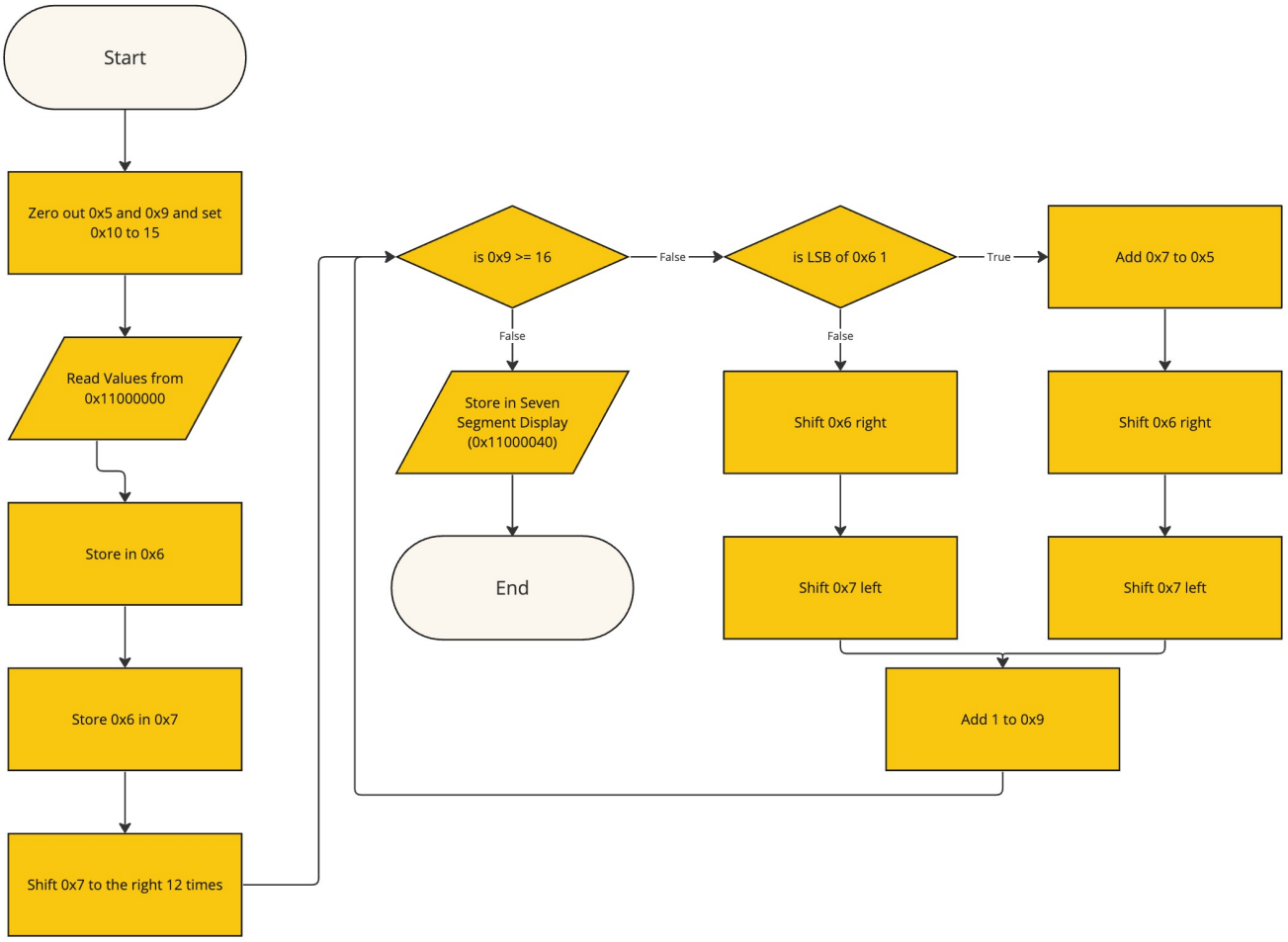

Figure 1: Multiplication Flowchart

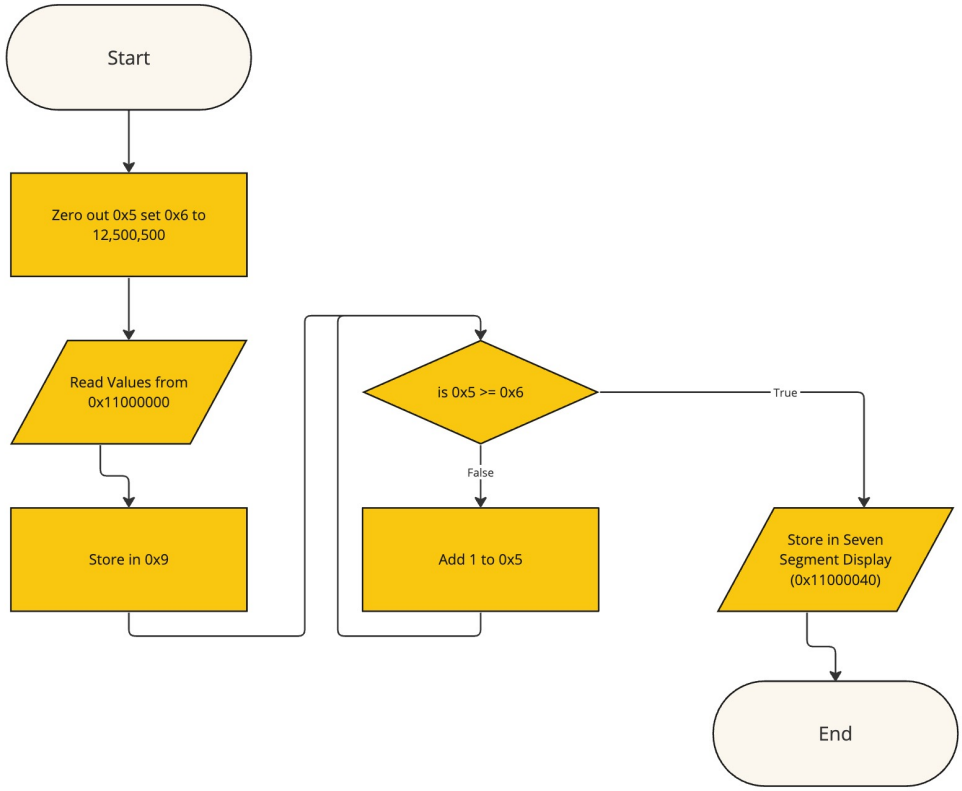

Figure 2: Instruction Delay Flowchart

2 Assembly Instructions

Below is the assembly instructions for each flow chart.

```
    # Setup registers
    lui     x5, 0x00000         # Output value from multiplication
    lui     x11, 0x11000        # Define Address for switch input
    lui     x9, 0x00000         # Current place counter
    addi    x10, x0, 0x00000010 # Number of places to multiply

    # Recive user input and format
    lw      x6, 0(x11)          # Load switch values to x6
    add     x7, x0, x6          # Load switch values to x7
    srli    x7, x7, 16          # Isolate second number

multiplyStep:
    # Begin multiplication
    bge     x9, x10, end        # If x9 >= x10 then end
    andi    x13, x6, 1          # Check is LSB is 1
    beqz    x13, isZero         # Branch if LSB is 0

    # Add shifted value to running total and then shift
    # registers and increment index
    add     x5, x5, x7
    srli    x6, x6, 1
    slli    x7, x7, 1
    addi    x9, x9, 1
    j       multiplyStep

isZero:
    # Shift registers and increment index
    srli    x6, x6, 1
    slli    x7, x7, 1
    addi    x9, x9, 1
    j       multiplyStep


end:
    sw      x5, 0x40(x11)       # Store final value
```

Listing 1: Assembly Code for Multiplication in Figure 1

```
    # Setup registers
    lui     x11, 0x11000    # Define Address for switch input
    lui     x5, 0x00000     # Output value from multiplication

    li      x6, 5000        # Define delay time by cycle count
    # Calculation of the cycle count:
    # Each instruction takes 40ns noting that there are 4
    # instructions in the count loop this means that one loop
    # takes 160ns. Divideing 0.5 seconds by the 160ns results
    # in 3,125,000 cycles.

    # Recive user input and format
    lw      x9, 0(x11)      # Load switch values to x9

countLoop:
    # Count up untill the desired time is met and the move
    # forward in the program
    bge     x5, x6, end
    addi    x5, x5, 1       # Increment loop counter
    nop
    j       countLoop

end:
    sw      x9, 0x40(x11)   # Store final value
```

Listing 2: Assembly Code for Delay in Figure 2

3 RARS Verification

Table 1: Flow Chart 1 Test Cases

Test Case	Input (0x5)	Expected Output	Actual Output
Zero Check	0x0000_0000	0x0000_0000	0x0000_0000
Alternating Check	0xaaaa_aaaa	0x71c6_38e4	0x71c6_38e4
Other Alternating Check	0x5555_5555	0x1c71_8e39	0x1c71_8e39
Max Check	0xffff_ffff	0xfffe0001	0xfffe0001

The test cases above demonstrate the code performs the desired outputs.
Test case 1 shows the zero case.
Test case 2 shows a value with alternating 1 and 0 .
Test case 3 shows a value with alternating 1 and 0 opposite of case 3.
Test case 5 shows an input at the maximum of the register.

Table 2: Flow Chart 2 Test Cases

Test Case	Input	Expected Output	Actual Output
Zero Check	0x0000_0000	0x0000_0000	0x0000_0000
Alternating Check	0xaaaa_aaaa	0xaaaa_aaaa	0xaaaa_aaaa
Other Alternating Check	0x5555_5555	0x5555_5555	0x5555_5555
Overflow Check	0xffff_ffff	0xffff_ffff	0xffff_ffff

The test cases above demonstrate the code performs the desired outputs.
Test case 1 shows the zero case.
Test case 2 shows a value with alternating 1 and 0 .
Test case 3 shows a value with alternating 1 and 0 opposite of case 3.
Test case 5 shows an input at the maximum of the register.