CPE 233 Software Assignment 2

Conditionals in Assembly

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

January 23, 2024

Table of Contents

1 Flow Charts

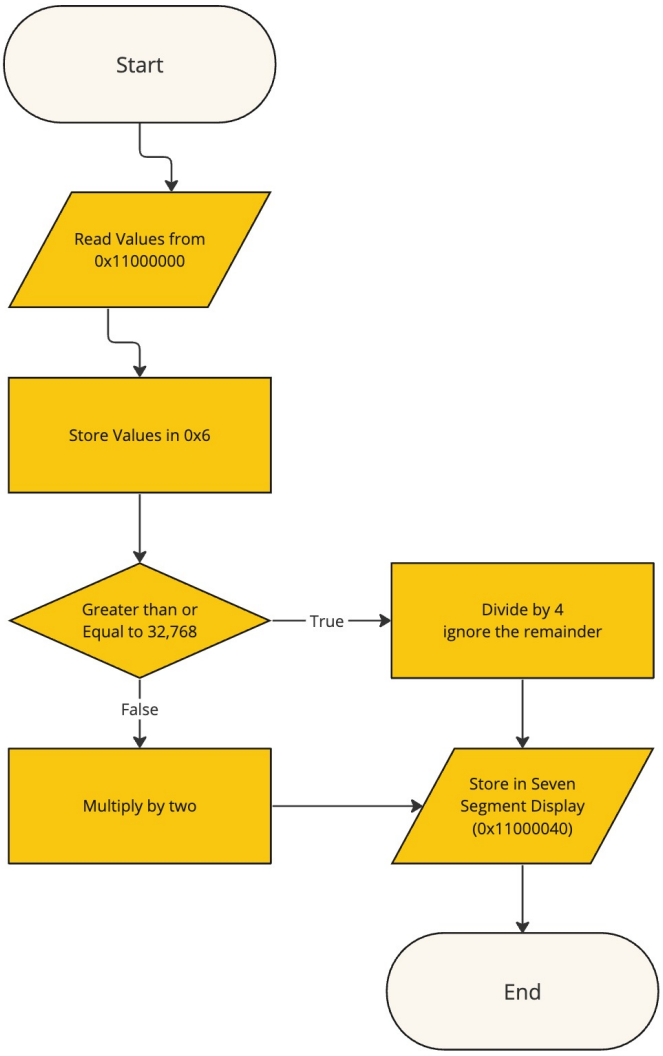

Figure 1: Single Condition Check

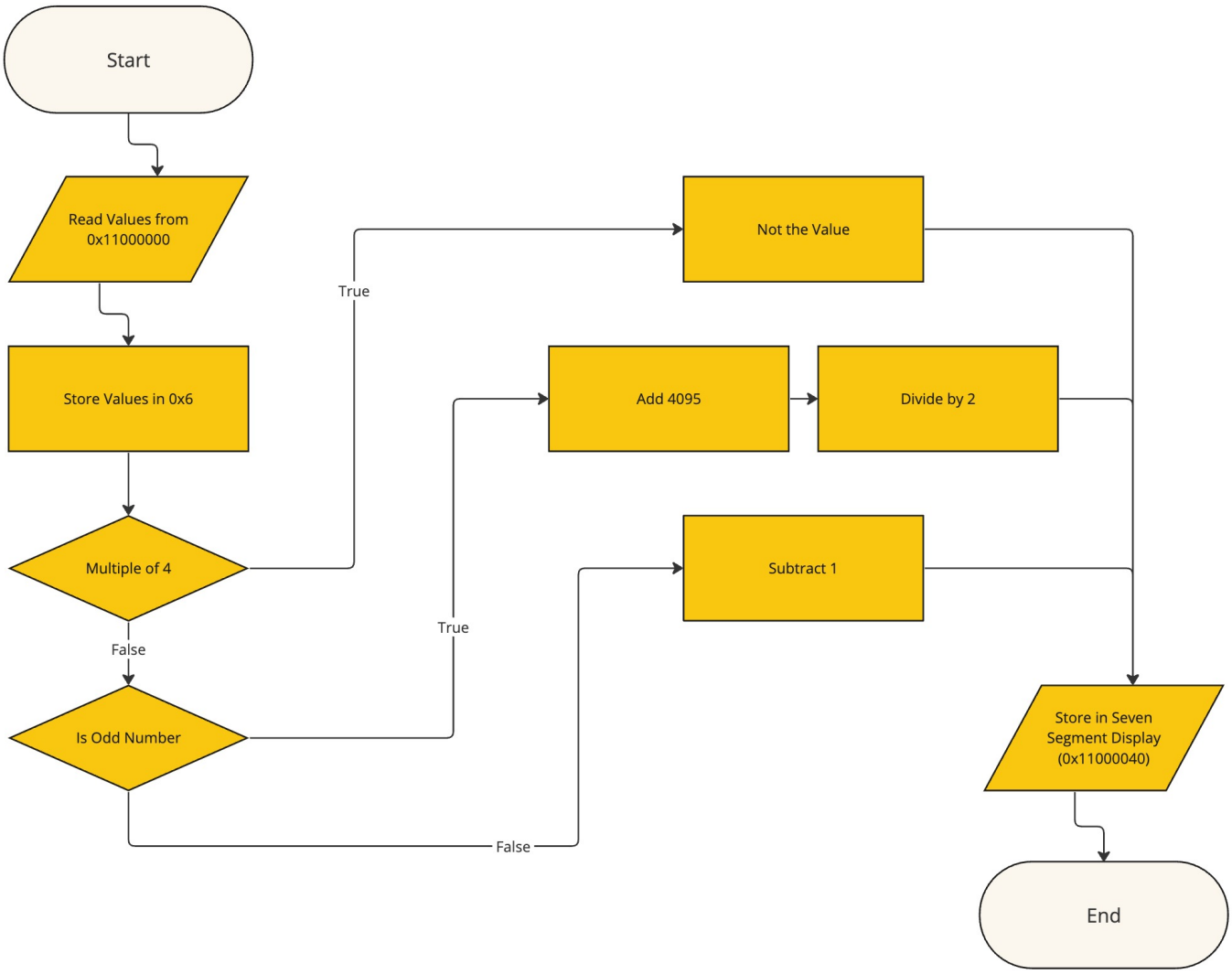

Figure 2: Multiple Condition Check

2 Assembly Instructions

Below is the assembly instructions for each flow chart.

```
    lui  x30, 0x11000
    lw   x5, 0(x30)          # load the switches in to memory
    li   x6, 0x00008000      # load the comparison number in to memory

    bge  x5, x6, isGreater   # check is number in switches is >= 32768
    slli x5, x5, 1           # multiply by 2 doing a shift 1 bit
    j    end

isGreater:
    srli x5, x5, 2           # divide by shifting 2 bits right

end:
    sw   x5, 0x40(x30)       # store to sseg
```

Listing 1: Assembly Code for Single Condition Check Figure 1

```
    lui  x30, 0x11000
    lw   x5, 0(x30)          # load the switches in to memory
    li   x6, 0xfffffffc      # or value for multiple of 4 check
    li   x7, 0xfffffffe      # or value for odd check
    li   x8, 4095            # value added if !multiple4 and odd

    or   x9, x5, x6          # or with x6 value to check for multiple4
    beq  x9, x6, isMultiple  # true if x5 is multiple of 4
    or   x9, x5, x7          # or with x7 value to check for odd
    bne  x9, x7, isOdd       # true if x5 is odd

    addi x5, x5, -1          # subtract 1 from !odd and !multiple4
    j    end

isMultiple:
    not  x5, x5              # store not of switches
    j    end
isOdd:
    add  x5, x5, x8
    srli x5, x5, 1           # divide by shifting 1 bits right

end:
    sw   x5, 0x40(x30)       # store to sseg
```

Listing 2: Assembly Code for Multiple Condition Check 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
Less Than Check	0x0000_2222	0x0000_4444	0x0000_4444
Equal to Check	0x0000_8000	0x0000_2000	0x0000_2000
Greater Than Check	0x00ab_ffff	0x002a_ffff	0x002a_ffff
Overflow Check	0xffff_ffff	0xffff_fffe	0xffff_fffe

The test cases above demonstrate the code performs the desired outputs.
Test case 1 shows the zero case.
Test case 2 show a value less than 32,768.
Test case 3 shows an input equal to 32,768.
Test case 4 shows an input greater than 32,768.
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	0xffff_ffff	0xffff_ffff
Multiple of 4 Check	0x0000_115c	0xffff_eea3	0xffff_eea3
Odd Check	0x0000_1159	0x0000_10ac	0x0000_10ac
!Odd !Multiple of 4 Check	0x0000_08ae	0x0000_08ad	0x0000_08ad
Overflow Check	0xffff_ffff	0x0000_07ff	0x0000_07ff

The test cases above demonstrate the code performs the desired outputs.
Test case 1 shows the zero case.
Test case 2 shows an input that is a multiple of 4.
Test case 3 shows an input that is an odd number.
Test case 4 shows an input that is not an odd number and is not a multiple of 4.
Test case 5 shows an input that is the maximum of the register.