**Design Requirements**

- Use Logisim to implement a set of instructions and the given circuit including a pipeline for fetching, decoding, and executing our instructions; include the design of a finite state machine in our control unit logic to drive the pipeline.
- Design a piece of software to embed using our instruction set.
- Use an 8051 microcontroller simulator to perform the exact same task as our machine’s embedded code; or some alternative assembly language execution (ARM, Motorola, etc.) microcontroller or microprocessor.
- Compare the performance of our machine to that of at least one existing microcontroller or microprocessor.

**Intel Assembly Code for Celsius to Fahrenheit**

```assembly
  .file "convert.c"
  .globl main
  .text
  .data
  .bss

main:
  .func main
  .type main, @function

  .comm main, gFunction
  .comm gFunction

  .section .rodata
  .string "Hello world!

  .section .text
  main:
    push ebp
    mov ebp, esp
    push ebx
    push ecx
    push edx

  find: loop
    mov eax, [ebp+arg0]
    shr eax, 1
    mov ebx, eax
    mov ecx, ebx
    mov edx, ebx
    cmp ecx, ebx
    jne find

  .size main, -.main
  .ident "GCC", "10.2.0"
  .section .note.GNU-stack,"",@prolog
```

We used the Intel x86 Assembly language to perform the task as our machine’s embedded code. Our program can successfully convert a Celsius value to Fahrenheit. Below are the opcodes of our program converted into hexadecimal for Logisim.

**ROM embedded Code for Celsius to Fahrenheit**

```
00000000 3a000009 3a010005 3a020020
38030000 14c40000 1c650000 05540000
00000000
```

**Computer Design Competition**

Tucker Strausbaugh
Sajid Amir
Kyler Killinger

EGR 433: Advanced Computer Engineering
Elizabethtown College, Elizabethtown, PA