ECED3403 – Assignment 4

Grace Yu

B00902046

July 12th, 2024

1. Design

1.1. Problem Introduction

This assignment aims to further develop the XM23p emulator. As of before assignment 4, the emulator can only perform sequential operations, one of the basic coding structures. The goal of assignment 4 is to implement two additional coding structures, conditionals and repetition. This will be accomplished by creating 9 new branching instructions, which modify the flow of control by modifying the program counter.

1.2. Design Section

PSEUDOCODE:

A small amount of code used or referenced in the pseudocode was documented in previous assignments or labs.

```
DECODE FUNCTION
   IF instruction opcode is BL
      SAVE 13 bit offset
   ELSE IF instruction opcode is between BEQ and BRA
      SAVE 10 bit offset
   END IF
END FUNCTION

EXECUTE1 FUNCTION
   SWITCH (opcode)
      CASE(BL)
         CALL bl_execute
         BREAK
      CASE(BEQ)
         CALL beq_execute
         BREAK
      CASE(BNE)
         CALL bne_execute
         BREAK
      CASE(BC)
         CALL bc_execute
         BREAK
      CASE(BNC)
         CALL bnc_execute
         BREAK
      CASE(BN)
         CALL bn_execute
         BREAK
      CASE(BGE)
         CALL bge_execute
         BREAK
      CASE(BLT)
         CALL blt_execute
         BREAK
      CASE(BRA)
         CALL bra_execute
         BREAK
   END SWITCH
END FUNCTIION

bl_execute FUNCTION
   SAVE PC to LR
   SAVE offset to PC
END FUNCTION

beq_execute FUNCTION
   PC <- Z = 1 ? PC + offset : PC
END FUNCTION

bne_execute FUNCTION
   PC <- Z = 0 ? PC + offset : PC
END FUNCTION

bc_execute FUNCTION
   PC <- C = 1 ? PC + offset : PC
END FUNCTION

bnc_execute FUNCTION
   PC <- C = 0 ? PC + offset : PC
END FUNCTION

bn_execute FUNCTION
   PC <- N = 1 ? PC + offset : PC
END FUNCTION

bge_execute FUNCTION
   PC <- (n ^ v) = 0 ? PC + offset : PC
END FUNCTION
```

```
blt_execute FUNCTION
   PC <- (n ^ v) = 1 ? PC + offset : PC
END FUNCTION

bra_execute
   PC <- PC + offset
END FUNCTION
```

1.3. Data Dictionary

```
offset = {16 [SET | CLEAR] 16}
SET = 1
CLEAR = 0

lr = {16 [SET | CLEAR] 16}
```