ECED3403 – Assignment 5

Grace Yu

B00902046

July 30th, 2024

1. Design

1.1. Problem Introduction

This assignment aims to further develop the XM23p emulator. The goal of assignment 4 is to implement the conditional execution instruction, CEX. Because branching can be expensive in an RISC, CEX can be used as an alternative option.

1.2. Design Section

PSEUDOCODE:

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

```
PIPELINE FUNCTION
…
IF odd clock tick
    IF bubble is TRUE
        cex_condition == OFF
        …
    END IF
…
END FUNCTION

DECODE FUNCTION
    IF instruction opcode is CEX
        SAVE condition
        SAVE cex_true
        SAVE cex_false
    END IF
END FUNCTION

cex_check FUNCTION
        IF cex_condition is TRUE and cex_true is greater than 0
        cex_true -= 1
        RETURN FALSE
    ELSE IF cex_condition is TRUE and cex_false is greater than 0
        cex_false -= 1
```

```
      RETURN TRUE
   ELSE IF cex_condition is FALSE and cex_false is greater than 0
      cex_true -= 1
      RETURN TRUE
   END IF
   ELSE IF cex_condition is FALSE and cex_true is greater than 0
      cex_true -= 1
      RETURN FALSE
   END IF
END FUNCTION

EXECUTE0 FUNCTION
   IF cex_condition is not OFF and CALL cex_check
      return
   END IF
   SWITCH (opcode)
      CASE(CEX)
         CALL cex_execute
         BREAK
   END SWITCH
END FUNCTION

cex_set FUNCTION
   IF condition is TRUE
      SET cex_condition to TRUE
   ELSE
      SET cex_condition to FALSE
   END IF
END FUNCTION

cex_execute FUNCTION
   SWITCH(code_suffix)
      CASE(EQ)
         CALL cex_condition with z == 1 condition
      BREAK
      CASE(NE)
         CALL cex_condition with z == 0 condition
      BREAK
      CASE(CSHS)
         CALL cex_condition with c == 1 condition
      BREAK
      CASE(CCLO)
         CALL cex_condition with c == 0 condition
      BREAK
      CASE(MI)
         CALL cex_condition with n == 1 condition
      BREAK
```

```
      CASE(PL)
         CALL cex_condition with n == 0 condition
      BREAK
      CASE(VS)
         CALL cex_condition with v == 1 condition
      BREAK
      CASE(VC)
         CALL cex_condition with v == 0 condition
      BREAK
      CASE(HI)
         CALL cex_condition with c == 1 and z == 0 condition
      BREAK
      CASE(LS)
         CALL cex_condition with c == 0 and z == 1 condition
      BREAK
      CASE(GE)
         CALL cex_condition with n == v condition
      BREAK
      CASE(LT)
         CALL cex_condition with n != v condition
      BREAK
      CASE(GT)
         CALL cex_condition with z == 0 and n == v condition
      BREAK
      CASE(LE)
         CALL cex_condition with z == 1 and n != v condition
      BREAK
      CASE(TR)
         CALL cex_condition with TRUE condition
      BREAK
      CASE(FL)
         CALL cex_condition with FALSE condition
      BREAK
   END SWITCH
END FUNCTION
```

1.3. Data Dictionary

```
cex_condition = [FALSE | TRUE | OFF]
FALSE = 0
TRUE = 1
OFF = 2

cex_true = {1 [SET | CLEAR] 8}
SET = 1
CLEAR = 0

cex_false = {0 [SET | CLEAR] 8}
```

```
code_suffix = [EQ | NE | CSHS | CCLO | MI | PL | VS | VC | HI | LS | GE | LT
| GT | LE | TR | FL]
EQ = 0
NE = 1
CSHS = 2
CCLO = 3
MI = 4
PL = 5
VS = 6
VC = 7
HI = 8
LS = 9
GE = 10
LT = 11
GT = 12
LE = 13
TR = 14
FL = 15

psw = v + n + z + c
v = [SET | CLEAR]
n = [SET | CLEAR]
z = [SET | CLEAR]
c = [SET | CLEAR]
```