ECED3403 – Lab 4

Grace Yu

B00902046

July 5th, 2024

# Design

## Problem Introduction

This lab aims to further develop the XM23p emulator. Lab 4 will implement the SETCC and CLRCC commands. SETCC will set specified flags, while CLRCC will clear specified flags. Additionally, debugging tools that display stages of the pipeline happening at every clock tick will be included. These two commands will allow for better debugging functionality, and the two new commands can be used in a variety of code applications.

## Design Section

**PSEUDOCODE:**

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

PIPELINE FUNCTION:

WHILE pc is not breakpoint AND instructionbit is not equal to 0

IF clock/2 remainder is equal to 0

IF DMAR is RD

CALL execute1

CALL fetch0

CALL decode0

ELSE

CALL fetch1

CALL execute0

END IF

IF increment mode is on AND clock/2 remainder is not equal to 0

return

END IF

INCREMENT CLOCK

END FUNCTION

DECODE FUNCTION

DEBUG

PRINT F0 and D0 stages

END DEBUG

IF instruction is SETCC or CLRCC

SAVE opcode

SAVE v, slp, n, z, c

. . . other code from previous assignments and labs

END IF

END FUNCTION

F1 FUNCTION

DEBUG

PRINT F1 and E0 stages

END DEBUG

. . . other code from previous assignments and labs

END FUNCTION

EXECUTE1 FUNCTION

SWITCH(opcode)

. . . other code from previous assignments and labs

CASE SETCC

CALL setcc\_execute

BREAK

CASE CLRCC

CALL clrcc\_execute

BREAK

END SWITCH

END FUNCTION

setcc\_execute FUNCTION

psw v |= operand v

psw slp |= operand slp

psw n |= operand n

psw z |= operand z

psw c |= operand c

END FUNCTION

clrcc\_execute FUNCTION

psw v &= ~operand v

psw slp &= ~operand slp

psw n &= ~operand n

psw z &= ~operand z

psw c &= ~operand c

END FUNCTION

execute1 FUNCTION

SWITCH(opcode)

CASE LD

CALL ld\_execute1

BREAK

CASE ST

CALL st\_execute1

BREAK

CASE LDR

CALL ldr\_execute1

BREAK

CASE STR

CALL str\_execute1

BREAK

END SWITCH

DCTRL = DONE

END FUNCTION

## Data Dictionary

operand = v + slp + n + z + c

v = [SET | CLEAR]

slp = [SET | CLEAR]

n = [SET | CLEAR]

z = [SET | CLEAR]

c = [SET | CLEAR]

SET = 1

CLEAR = 1