ECED3403 – Lab 2

Grace Yu

B00902046

May 31st, 2024

# Design

## Problem Introduction

This lab aims to further develop the XM23p emulator that was begun in assignment 1. A crucial component of the three-stage pipeline is the decode stage. The program completed in this lab will successfully fetch and decode a specified 22 out of the 40 commands that XM23p recognizes.

## Design Section

Pseudocode:

FUNCTION pipeline:

SET PC to S9 record value or 0

SET CLOCK to 0

SET ICNTRL to 0

IF CLOCK value is even THEN

CALL F0

CALL D0

ELSE

CALL F1

CALL E0

END IF

CLOCK++

END FUNCTION

FUNCTION F0

SET instructionaddress to PC

INCREMENT PC by 2

SET ictrl to READ

END FUNCTION

FUNCTION F1

FETCH two byte long instructionbit from imem array

END FUNCTION

FUNCTION D0

IF instructionbit is between LDR and STR

instruction is not part of A2

ELSE IF instructionbit is between BL and BRA

instruction is not part of A2

ELSE IF instructionbit is between MOVL and MOVH

SET arrayplace to instructionbit

MASK arrayplace except for bits 12, 11

SHIFT arrayplace to the right by 11

SET insturctionmnem to MOVL + arrayplace

SET bytevalue to CALL savebytevalue

ELSE IF instructionbit is between LD and ST

instruction is not part of A2

ELSE IF instructionbit is between MOV and CLRCC

IF instructionbit is between SETPRI to CLRCC

instruction is not part of A2

ELSE instruction is between MOV and SXT

IF instructionbit is between MOV and SWAP

SET arrayplace to instructionbit

MASK arrayplace except for bits 7

SHIFT arrayplace to the right by 7

SET intructionmnem to MOV + arrayplace

SET sourceconstant to CALL savesourceconstant

ELSE instruction is between SRA and SXT

IF instructionbit is between SRA and RRC

SET arrayplace to instructionbit

MASK arrayplace except for bits 5, 4, 3

SHIFT arrayplace to the right by 3

SET insturctionmnem to SRA + arrayplace

ELSE instructionbit is between SWPB and SXT

SET arrayplace to instructionbit

MASK arrayplace except for bits 5

SHIFT arrayplace to the right by 5

SET insturctionmnem to SWPB + arrayplace

END IF

END IF

SET wordbyte to CALL savewordbyte

END IF

ELSE instructionbit is between ADD and BIS

SET arrayplace to instructionbit

MASK arrayplace except for bits 11, 10, 9, 8

SHIFT arrayplace to the right by 8

SET insturctionmnem to ADD + arrayplace

SET wordbyte to CALL savewordbyte

SET sourceconstantcheck to CALL savesourceconstantcheck

SET sourceconstant to CALL savesourceconstant

END IF

CALL printdecode

END FUNCTION

FUNCTION savesourceconstant

MASK instructionbit except for bits 5, 4, 3

SHIFT instructionbit by 3

SET sourceconstant to constantarray[instructionbit]

RETURN sourceconstant

END FUNCTION

FUNCTION savewordbyte

MASK instructionbit except for bits 5, 4, 3

SHIFT instructionbit by 3

SET sourceconstant to constantarray[instructionbit]

RETURN wordbyte

END FUNCTION

FUNCTION savebytevalue

MASK instructionbit except for bits 10, 9, 8, 7, 6, 5, 4, 3

SHIFT instructionbit by 3

SET sourceconstant to constantarray[instructionbit]

RETURN bytevalue

END FUNCTION

FUNCTION printdecode

IF instruction is part of A2

PRINT instructionaddress and mnemarray[instructionmnem]

ELSE

PRINT instructionaddress and instructionbit

END IF

IF instructionmnem is between ADD and BIS

PRINT sourceconstantcheck

END IF

IF instructionmnem is between ADD and MOV, OR instructionmnem is between

SRA and RRC

print wordbyte

END IF

IF instructionmnem is between ADD and SWAP

IF sourceconstantcheck is 0, OR instructionmnem is MOV, or

instrutionmnem is SWAP

PRINT source

ELSE

PRINT constant

END IF

PRINT destination

END FUNCTION

data dictionary

constant array

pre condition  
ir   
address

post condition:  
register-constant  
WB  
RC  
SRC  
DST

struct reg.con{

int WB

int RC

int SRC

int DST

}

struct movx{

char byte

int dest

}

in e0:

switch case (check mnemonic)

case ADD

look at regcon structure

case MOVX

look at movx structure

enum codes code //decoded instruction

union d2r\_info decoded //decoded instruction info

// both global

## Data Dictionary

CLOCK

PC

IMEM

ICNTRL

IMAR