|  |
| --- |
|  |
| Testing Plan |
| WI11 Machine Program |
| **Elliot Schumacher** |
| **Brad Kline, Zach Smith, Ryan Powers, and Evan Todd** |
| **3/9/2011** |

Contents

[Introduction 3](#_Toc287577344)

[Assembly Files 3](#_Toc287577345)

[Relocatable 3](#_Toc287577346)

[Non-Relocatable 3](#_Toc287577347)

[Multiple Assembly Files 6](#_Toc287577348)

[First Example 6](#_Toc287577349)

[Second Example 7](#_Toc287577350)

# Introduction

This testing document shows the testing processes performed on the WI11 Machine software suite. Also included are individual testing plans for each component; these are more comprehensive than this documents. This testing plan is to document that the software works as a suite.

# Assembly Files

## Relocatable

;Test program for WI11 suite (relocatable)

Test1 .ORIG

HALT .EQU x25

EG0 .EQU x0

SPACE .BLKW x1

COUNT .FILL x5

BEGIN TRAP x43

ST EG0,SPACE

AND R1,R1,x10

NOT R2,R1

LD R3,SPACE

ADD R4,R2,R3

LD R5,COUNT

ADD R5,R5,#-1

ST R5,COUNT

BRNP BEGIN ;If CCRs are not sent to Zero, will branch back to BEGIN

ADD EG0,R4,x0

TRAP x31

TRAP HALT

.END

This program should produce the following file when run through the assembler.

## Non-Relocatable

### First Example

; Sample program

Test2 .ORIG x30B0

count .FILL #4

PUTS .EQU x22

ANS .BLKW x1

QUEST .STRZ "Which company introduced the first laptop computer in 1981?"

ANS1 .STRZ "a) Hewlett Packard"

ANS2 .STRZ "b) Dell"

ANS3 .STRZ "c) Epson"

ANS4 .STRZ "d) Compaq"

INCO .STRZ "Incorrect, the correct answer is:"

CORRA .STRZ "That's Correct!"

Begin LEA R0,QUEST

TRAP PUTS

LEA R0,ANS1

TRAP PUTS

LEA R0,ANS2

TRAP PUTS

LEA R0,ANS3

TRAP PUTS

LEA R0,ANS4

TRAP PUTS

TRAP x23

LD R1,CORR

ADD R2,R1,R0

NOT R2,R2

BRZ ESUL

LEA R0,INCO

TRAP PUTS

LEA R0,ANS3

TRAP PUTS

EXIT TRAP HALT

ESUL LEA R0,CORRA

TRAP PUTS

BRNZP EXIT

; ----- Scratch Space -----

Array .BLKW #3

CORR .FILL xFF9C

MASK .FILL xFFFF

.END Begin

HALT .EQU x25

### Second Example

; Sample program

Test3 .ORIG x30B0

INN .EQU x33

OUTN .EQU x31

PUTS .EQU x22

PROMPT .STRZ "Enter 'a' for add, 's' for subtract, or 'm' for mulitply"

PRMPT1 .STRZ "Enter the first number to be added:"

PRMPT2 .STRZ "Enter the number to be subtracted from"

PRMPT3 .STRZ "Enter the first number to multiply:"

PRMPT4 .STRZ "Enter the second number to be added:"

PRMPT5 .STRZ "Enter the number to subtact:"

PRMPT6 .STRZ "Enter second number to multiply"

Begin LEA R0,PROMPT

TRAP PUTS

TRAP x23

LD R1,CHARA

ADD R2,R1,R0

NOT R2,R2

BRZ OPADD

LD R1,CHARS

ADD R2,R1,R0

NOT R2,R2

BRZ OPSUB

LD R1,CHARM

ADD R2,R1,R0

NOT R2,R2

BRZ OPMUL

EXIT TRAP HALT

OPADD LEA R0,PRMPT1

TRAP PUTS

TRAP INN

ST R0,TEMP

LEA R0,PRMPT4

TRAP PUTS

TRAP INN

LD R1,TEMP

ADD R0,R0,R1

TRAP OUTN

BRNZP EXIT

OPSUB LEA R0,PRMPT2

TRAP PUTS

TRAP INN

ST R0,TEMP

LEA R0,PRMPT5

TRAP PUTS

TRAP INN

LD R1,TEMP

NOT R0,R0

ADD R0,R0,R1

ADD R0,R0,#1

TRAP OUTN

BRNZP EXIT

OPMUL LEA R0,PRMPT3

TRAP PUTS

TRAP INN

ST R0,TEMP

LEA R0,PRMPT6

TRAP PUTS

TRAP INN

ST R0,TEMP2

LD R2,TEMP2

LD R1,TEMP

ADD R1,R1,#-1

MUL ADD R0,R0,R2

ADD R1,R1,#-1

BRNP MUL

TRAP OUTN

BRNZP EXIT

; ----- Scratch Space -----

TEMP .BLKW #1

TEMP2 .BLKW #1

CHARA .FILL xFF9E

CHARS .FILL xFF8C

CHARM .FILL xFF92

.END Begin

HALT .EQU x25

### Third Example

;Example non-relocatable program for WI11 program.

Test4 .ORIG x30B0

count .FILL #4

Begin LD ACC,count ;R1 <- 4

LEA R0,msg

loop TRAP x22 ;print "hi! "

ADD ACC,ACC,#-1 ;R1--

BRP loop

JMP Next

msg .STRZ "hi! "

Next AND R0,R0,x0 ;R0 <- 0

NOT R0,R0 ;R0 <- xFFFF

ST R0,Array ;M[Array] <- xFFFF

LEA R5,Array

LD R6,=#100 ;R6 <= #100

STR R0,R5,#1 ;M[Array+1] <= xFFFF

TRAP x25

ACC .EQU #1

; ----- Scratch Space -----

Array .BLKW #3

.FILL x10

.END Begin

## Multiple Assembly Files

### First Example

;Test program for WI11 suite

Test0 .ORIG

.EXT GENER

STORA1 .BLKW x1

STORA2 .BLKW x1

COUNT .FILL x2

OUTN .EQU x31

HALT .EQU x25

BEGIN JSR GENER

ST R0,STORA1

JSR GENER

ST R0,STORA2

LD R1,STORA1

LD R2,STORA2

ADD R0,R1,R2

TRAP OUTN

TRAP HALT

.END

;Test program for WI11 suite (relocatable)

Test1 .ORIG

.ENT GENER

SPACE .BLKW x1

HALT .EQU x25

EG0 .EQU x0

COUNT .FILL x5

GENER TRAP x43

ST EG0,SPACE

AND R1,R1,x10

NOT R2,R1

LD R3,SPACE

ADD R4,R2,R3

LD R5,COUNT

ADD R5,R5,#-1

ST R5,COUNT

BRNP GENER ;If CCRs are not sent to Zero, will branch back to BEGIN

ADD EG0,R4,x0

RET

TRAP HALT

.END

### Second Example

;Test Program for WI11

;Provided by instructor

;234567890123456789012345678901234567890

;label\_\_\_opppp\_\_\_operandsandcomments...

;

Main .ORIG

.EXT Displ,V

.ENT Start

.EXT X

;

Start JSR Displ ;Display 6..0

LD R1,V ;r1 <- M[V]

ST R1,X ;M[X] <- r1

JSR Displ ;Display 2..0

TRAP x25 ;halt

.END Start

;Subroutine for displaying a series of lines of text

; The lines of text display a count-down, from X to 0

;Calling convention: register 3 contains return address

;Provided by instructor

;234567890123456789012345678901234567890

;label\_\_\_opppp\_\_\_operandsandcomments...

;

Mesg .ORIG

.ENT Displ,X

;

Txt .STRZ "Value= "

X .FILL #6

SavR0 .BLKW #1

SavR1 .BLKW #1

SavR7 .BLKW #1

;

Displ ST R0,SavR0 ;save reg that will be over-written

ST R1,SavR1

ST R7,SavR7

LD R1,X ;r1 <- M[X]

BRN Done ;if (r1 < 0) goto Done

Loop LEA R0,Txt

TRAP x22 ;Display text "Value= "

LD R0,X

TRAP x31 ;Display value in M[X]

ADD R0,R0,#-1

ST R0,X ;M[X] <- r0

BRN Done ;if (r0 < 0) goto Done

JMP Loop ;goto Loop

Done LD R0,SavR0 ;restore registers

LD R1,SavR1

LD R7,SavR7

RET

.END Displ

;234567890123456789012345678901234567890

;label\_\_\_opppp\_\_\_operandsandcomments...

;Provided by instructor

;

Data .ORIG

.EXT X

.ENT V

V .FILL #2

TRAP x43

Done TRAP x25

LD R1,=#1

.END Done

## Errors

### First Example

; Assembler should give an error; the program spans multiple memory pages

Test6 .ORIG x0FF0

count .FILL #4

Begin LD ACC,count ;R1 <- 4

LEA R0,msg

loop TRAP x22 ;print "hi! "

ADD ACC,ACC,#-1 ;R1--

BRP loop

JMP Next

msg .STRZ "hi! "

Next AND R0,R0,x0 ;R0 <- 0

NOT R0,R0 ;R0 <- xFFFF

ST R0,Array ;M[Array] <- x####

LEA R5,Array

LD R6,=#100 ;R6 <= #100

STR R0,R5,#1 ;M[Array+1] <= x####

TRAP x25

ACC .EQU #1

; ----- Scratch Space -----

Array .BLKW #3

.FILL x10

.END Begin