|  |
| --- |
|  |
| 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.

To keep this guide short, not all files produced by the assembler and linker are included. Samples from relevant cases are included.

# 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.

HTest1 MMMM000f

T00010005

T0002f043

T00033000M0

T00045270

T00059440

T00062600M0

T00071883

T00082a01M0

T00091b7f

T000a3a01M0

T000b0a02M0

T000c1120

T000df031

T000ef025

E0000

This program should produce the following file when run through the linker. (The address x3000 was assigned to it manually).

HTest1 3000000f

T30010005

T3002f043

T30033000

T30045270

T30059440

T30062600

T30071883

T30082a01

T30091b7f

T300a3a01

T300b0a02

T300c1120

T300df031

T300ef025

E3000

## 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

When run through the assembler, the following is produced.

HTest2 30b000ba

T30b00004

T30b20057

T30b30068

T30b40069

T30b50063

T30b60068

T30b70020

T30b80063

T30b9006f

T30ba006d

T30bb0070

T30bc0061

T30bd006e

T30be0079

T30bf0020

T30c00069

T30c1006e

T30c20074

T30c30072

T30c4006f

T30c50064

T30c60075

T30c70063

T30c80065

T30c90064

T30ca0020

T30cb0074

T30cc0068

T30cd0065

T30ce0020

T30cf0066

T30d00069

T30d10072

T30d20073

T30d30074

T30d40020

T30d5006c

T30d60061

T30d70070

T30d80074

T30d9006f

T30da0070

T30db0020

T30dc0063

T30dd006f

T30de006d

T30df0070

T30e00075

T30e10074

T30e20065

T30e30072

T30e40020

T30e50069

T30e6006e

T30e70020

T30e80031

T30e90039

T30ea0038

T30eb0031

T30ec003f

T30ed0000

T30ee0061

T30ef0029

T30f00020

T30f10048

T30f20065

T30f30077

T30f4006c

T30f50065

T30f60074

T30f70074

T30f80020

T30f90050

T30fa0061

T30fb0063

T30fc006b

T30fd0061

T30fe0072

T30ff0064

T31000000

T31010062

T31020029

T31030020

T31040044

T31050065

T3106006c

T3107006c

T31080000

T31090063

T310a0029

T310b0020

T310c0045

T310d0070

T310e0073

T310f006f

T3110006e

T31110000

T31120064

T31130029

T31140020

T31150043

T3116006f

T3117006d

T31180070

T31190061

T311a0071

T311b0000

T311c0049

T311d006e

T311e0063

T311f006f

T31200072

T31210072

T31220065

T31230063

T31240074

T3125002c

T31260020

T31270074

T31280068

T31290065

T312a0020

T312b0063

T312c006f

T312d0072

T312e0072

T312f0065

T31300063

T31310074

T31320020

T31330061

T3134006e

T31350073

T31360077

T31370065

T31380072

T31390020

T313a0069

T313b0073

T313c003a

T313d0000

T313e0054

T313f0068

T31400061

T31410074

T31420027

T31430073

T31440020

T31450043

T3146006f

T31470072

T31480072

T31490065

T314a0063

T314b0074

T314c0021

T314d0000

T314ee0b2

T314ff022

T3150e0ee

T3151f022

T3152e101

T3153f022

T3154e109

T3155f022

T3156e112

T3157f022

T3158f023

T31592368

T315a1440

T315b9480

T315c0562

T315de11c

T315ef022

T315fe109

T3160f022

T3161f025

T3162e13e

T3163f022

T31640f61

T3168ff9c

T3169ffff

E314e

The file produced by the linker is similar, and therefore not included.

### 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

The produced files are not included, as the output is similar in form to the above example.

### 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

The produced files are not included, as the output is similar in form to the above example.

## 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

The two output files were produced (they are in order).

HTest0 MMMM0009

I000380GENER

I000580GENER

T00020002

T00034800

T00041220

T00054800

T00061040

T0007f031

T0008f025

E0000

HTest1 MMMM0010

XR0000GENER

T00020005

T0003f043

T00043001M0

T00055270

T00069440

T00072601M0

T00081883

T00092a02M0

T000a1b7f

T000b3a02M0

T000c0a03M0

T000d1120

T000ed000

T000ff025

E0000

The object file produced by the linker is as follows (the address x0 was supplied by the user in the command call).

HTest0 00000019

T00020002

T00034809

T00041220

T00054809

T00061040

T0007f031

T0008f025

T000b0005

T000cf043

T000d300a

T000e5270

T000f9440

T0010260a

T00111883

T00122a0b

T00131b7f

T00143a0b

T00150a0c

T00161120

T0017d000

T0018f025

E0000

### 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

As the example output and object files are similar to the first example, they are not included.