

CSE30321 Computer Architecture I

A Short Tutorial on XSPIM

XSPIM is a software that simulates the execution of MIPS assembly programs. It does a context and syntax check while loading an assembly program. In addition, it adds in necessary overhead instructions as it sees fit and keeps track of all the register and memory contents as each instruction is executed. Below, we give a quick summary on the usage of XSPIM.

- Go to the directory where your assembly language program is stored.
- Type `/afs/nd.edu/user14/csesoft/bin/xspim` at the prompt. (Adding the full directory to your path or making an alias for `xspim` will ease future invocations.)
- A window pops up as shown in Figure 1. The window is divided into five sections vertically.
 - Section 1: displays all register contents. You only need to get familiar with the registers which we have discussed in class.
 - Section 2: is the *Command Console*. Each button here corresponds to a command supported by the simulator.
 - Section 3: is the *Text Segment* which displays the MIPS instructions loaded into the memory and to be executed. From left to right, we have the memory addresses, the corresponding memory contents in hex, the actual MIPS instructions, and the corresponding assembly instructions.
 - Section 4: is the *Data Segment* which displays memory addresses and their contents in the data and stack segments of the memory.
 - Section 5: is the *Information Console* which shows the actions taken by the simulator.
- The functions of the command buttons in the *Command Console* are summarized below:
 - quit** Exit from the simulator.
 - load** Read an assembly program file into memory.
 - reload** Reload the assembly program file.
 - run** Execute a program to completion.
 - step** Advance the execution of a program by a given step size.
 - clear** Reinitialize registers or memory. (You have a choice of what to clear. To reload a program you modified, you need to clear both the registers and the memory.)
 - setvalue** Set the value in register or memory.
 - print** Print the value in register or memory.
 - breakpoint** Set or delete a breakpoint.
 - help** Display the above message.

We will not use *terminal* and *mode* in this lab.

xspim

PC = 00400000 EPC = 00000000 Cause = 00000000 BadVAddr= 00000000
 Status = 00000000 HI = 00000000 LO = 00000000

General Registers

R0 (r0) = 00000000	R8 (t0) = 00000000	R16 (s0) = 00000000	R24 (t8) = 00000000
R1 (a0) = 00000000	R9 (t1) = 00000000	R17 (s1) = 00000000	R25 (t9) = 00000000
R2 (v0) = 00000000	R10 (t2) = 00000000	R18 (s2) = 00000000	R26 (k0) = 00000000
R3 (v1) = 00000000	R11 (t3) = 00000000	R19 (s3) = 00000000	R27 (k1) = 00000000
R4 (a0) = 00000000	R12 (t4) = 00000000	R20 (s4) = 00000000	R28 (gp) = 10008000
R5 (a1) = 00000000	R13 (t5) = 00000000	R21 (s5) = 00000000	R29 (sp) = 7ffffeffc
R6 (a2) = 00000000	R14 (t6) = 00000000	R22 (s6) = 00000000	R30 (s8) = 00000000
R7 (a3) = 00000000	R15 (t7) = 00000000	R23 (s7) = 00000000	R31 (ra) = 00000000

Double Floating Point Registers

FP0 = 0.00000	FP8 = 0.00000	FP16 = 0.00000	FP24 = 0.00000
FP2 = 0.00000	FP10 = 0.00000	FP18 = 0.00000	FP26 = 0.00000
FP4 = 0.00000	FP12 = 0.00000	FP20 = 0.00000	FP28 = 0.00000
FP6 = 0.00000	FP14 = 0.00000	FP22 = 0.00000	FP30 = 0.00000

Single Floating Point Registers

FP0 = 0.00000	FP8 = 0.00000	FP16 = 0.00000	FP24 = 0.00000
---------------	---------------	----------------	----------------

quit load reload run step clear

set value print breakpoints help terminal mode

Text Segments

[0x00400000]	0x8fa40000	lw \$4, 0(\$29)	; 102: lw \$a0, 0(\$sp)
[0x00400004]	0x27a50004	addiu \$5, \$29, 4	; 103: addiu \$a1, \$sp, 4
[0x00400008]	0x24a60004	addiu \$6, \$5, 4	; 104: addiu \$a2, \$a1, 4
[0x0040000c]	0x00041080	sll \$2, \$4, 2	; 105: sll \$v0, \$a0, 2
[0x00400010]	0x00c23021	addu \$6, \$6, \$2	; 106: addu \$a2, \$a2, \$v0
[0x00400014]	0x0c000000	jal 0x00000000 [main]	; 107: jal main
[0x00400018]	0x3402000a	ori \$2, \$0, 10	; 108: li \$v0 10
[0x0040001c]	0x0000000c	syscall	; 109: syscall

KERNEL

Data Segments

DATA					
[0x10000000]...	[0x10020000]	0x00000000			
STACK					
[0x7ffffeffc]		0x00000000			
KERNEL DATA					
[0x90000000]		0x20204578	0x63657074	0x696f6e20	0x00206f63
[0x90000010]		0x63757272	0x65642061	0x6e642069	0x676e6f72

SPIM Version 6.2 of January 11, 1999
 Copyright 1990-1998 by James R. Larus (larus@cs.wisc.edu).
 All Rights Reserved.
 See the file README for a full copyright notice.
 Loaded: /usr/local/src/spim/bin//trap.handler

Figure 1: XSPIM Window