

Diep (Emma) Vu

Lab 5 Report

1. Change of registers in sum.asm

QtSpim

FP Regs	Int Regs [10]
	Int Regs [10]
PC = 4194360 EPC = 4194356 Cause = 36 BadAddr = 0 Status = 805371664 HI = 0 LO = 0 R0 [r0] = 0 R1 [at] = 268500992 R2 [v0] = 4 R3 [v1] = 0 R4 [a0] = 1 R5 [a1] = 2147483892 R6 [a2] = 2147483100 R7 [a3] = 0 R8 [t0] = -35 R9 [t1] = 0 R10 [t2] = 0 R11 [t3] = 0 R12 [t4] = 0 R13 [t5] = 0 R14 [t6] = 0 R15 [t7] = 0 R16 [s0] = 0 R17 [s1] = 0 R18 [s2] = 0 R19 [s3] = 0 R20 [s4] = 0 R21 [s5] = 0 R22 [s6] = 0 R23 [s7] = 0 R24 [t8] = 0 R25 [t9] = 0 R26 [t10] = 0 R27 [k1] = 0 R28 [gp] = 268468224 R29 [sp] = 2147483088 R30 [s8] = 0 R31 [ra] = 4194328	User Text Segment [00400000]..[00440000] ; 183: lw \$a0 \$0(\$sp) # argc ; 184: addiu \$a1 \$sp 4 # argv ; 185: addiu \$a2 \$a1 4 # envp ; 186: addiu \$a3 \$a2 2 ; 187: addiu \$a2 \$a2 \$v0 ; 188: jal main ; 189: nop ; 190: li \$v0 10 ; 192: syscall # syscall 10 (exit) ; 193: lw \$t0, num1 # temp = num1 ; 194: add \$t1, \$0, \$t0 # accum = temp ; 20: lw \$t0, num2 # temp = num2 ; 19: add \$t1, \$0, \$t0 # accum = temp ; 20: lw \$t0, num2 # temp = num2 Breakpoint 1, \$t0 # accum = accum + temp # temp = num3 Execution stopped at breakpoint at 0x0040 I, \$t0 # accum = accum + temp sum = accum ; 90: move \$k1 \$at # Save Sat ; 92: sw \$v0 \$1 # Not re-entrant and we can't trust \$sp ; 93: sw \$a0 \$2 # But we need to use these registers ; 95: mfcb \$k0 \$13 # Cause register ; 96: srl \$a0 \$k0 2 # Extract ExcCode Field ; 97: andi \$a0 \$a0 0x1f ; 101: li \$v0 4 # syscall 4 (print_int) ; 102: la \$a0 __m1_ ; 103: syscall ; 104: srl \$a0 1 # syscall 1 (print_int) ; 106: srl \$a0 \$k0 2 # Extract ExcCode Field ; 107: andi \$a0 \$a0 0x1f ; 108: syscall ; 110: li \$v0 4 # syscall 4 (print_int) ; 111: andi \$a0 \$k0 0x3c ; 112: lw \$a0 __excp(\$a0) ; 113: nop ; 114: syscall ; 116: bnez \$a0 __auto_no_no # And DC exception requires special checks Kernel Text Segment [80000000]..[80010000] ; 99: move \$k1 \$at # Save Sat ; 100: mfcb \$k0 \$13 # Cause register ; 101: srl \$a0 \$k0 2 # Extract ExcCode Field ; 102: andi \$a0 \$a0 0x1f ; 103: syscall ; 104: srl \$a0 1 # syscall 1 (print_int) ; 106: srl \$a0 \$k0 2 # Extract ExcCode Field ; 107: andi \$a0 \$a0 0x1f ; 108: syscall ; 109: li \$v0 4 # syscall 4 (print_int) ; 110: andi \$a0 \$k0 0x3c ; 111: lw \$a0 __excp(\$a0) ; 112: bnez \$a0 __auto_no_no # And DC exception requires special checks

All Rights Reserved.
SPIM is distributed under a BSD license.
See the file README for a full copyright notice.
QtSPIM is linked to the Qt library, which is distributed under the GNU Lesser General Public License version 3 and version 2.1.

QtSpim

FP Regs	Int Regs [10]
	Int Regs [10]
PC = 4194372 EPC = 4194368 Cause = 36 BadAddr = 0 Status = 805371664 HI = 0 LO = 0 R0 [r0] = 0 R1 [at] = 268500992 R2 [v0] = 4 R3 [v1] = 0 R4 [a0] = 1 R5 [a1] = 2147483892 R6 [a2] = 2147483100 R7 [a3] = 0 R8 [t0] = -276 R9 [t1] = -18 R10 [t2] = 0 R11 [t3] = 0 R12 [t4] = 0 R13 [t5] = 0 R14 [t6] = 0 R15 [t7] = 0 R16 [s0] = 0 R17 [s1] = 0 R18 [s2] = 0 R19 [s3] = 0 R20 [s4] = 0 R21 [s5] = 0 R22 [s6] = 0 R23 [s7] = 0 R24 [t8] = 0 R25 [t9] = 0 R26 [t10] = 0 R27 [k1] = 0 R28 [gp] = 268468224 R29 [sp] = 2147483088 R30 [s8] = 0 R31 [ra] = 4194328	User Text Segment [00400000]..[00440000] ; 183: lw \$a0 \$0(\$sp) # argc ; 184: addiu \$a1 \$sp 4 # argv ; 185: addiu \$a2 \$a1 4 # envp ; 186: addiu \$a3 \$a2 2 ; 187: addiu \$a2 \$a2 \$v0 ; 188: jal main ; 189: nop ; 190: li \$v0 10 ; 192: syscall # syscall 10 (exit) ; 18: lw \$t0, num1 # temp = num1 ; 19: add \$t1, \$0, \$t0 # accum = temp ; 20: lw \$t0, num2 # temp = num2 ; 19: add \$t1, \$0, \$t0 # accum = temp ; 20: lw \$t0, num2 # temp = num2 Breakpoint 1, \$t0 # accum = accum + temp # temp = num3 Execution stopped at breakpoint at 0x0040 I, \$t0 # accum = accum + temp sum = accum ; 90: move \$k1 \$at # Save Sat ; 92: sw \$v0 \$1 # Not re-entrant and we can't trust \$sp ; 93: sw \$a0 \$2 # But we need to use these registers ; 95: mfcb \$k0 \$13 # Cause register ; 96: srl \$a0 \$k0 2 # Extract ExcCode Field ; 97: andi \$a0 \$a0 0x1f ; 101: li \$v0 4 # syscall 4 (print_int) ; 102: la \$a0 __m1_ ; 103: syscall ; 104: srl \$a0 1 # syscall 1 (print_int) ; 106: srl \$a0 \$k0 2 # Extract ExcCode Field ; 107: andi \$a0 \$a0 0x1f ; 108: syscall ; 109: li \$v0 4 # syscall 4 (print_int) ; 110: andi \$a0 \$k0 0x3c ; 111: lw \$a0 __excp(\$a0) ; 112: bnez \$a0 __auto_no_no # And DC exception requires special checks Kernel Text Segment [80000000]..[80010000] ; 99: move \$k1 \$at # Save Sat ; 100: mfcb \$k0 \$13 # Cause register ; 101: srl \$a0 \$k0 2 # Extract ExcCode Field ; 102: andi \$a0 \$a0 0x1f ; 103: syscall ; 104: srl \$a0 1 # syscall 1 (print_int) ; 106: srl \$a0 \$k0 2 # Extract ExcCode Field ; 107: andi \$a0 \$a0 0x1f ; 108: syscall ; 109: li \$v0 4 # syscall 4 (print_int) ; 110: andi \$a0 \$k0 0x3c ; 111: lw \$a0 __excp(\$a0) ; 112: bnez \$a0 __auto_no_no # And DC exception requires special checks

All Rights Reserved.
SPIM is distributed under a BSD license.
See the file README for a full copyright notice.
QtSPIM is linked to the Qt library, which is distributed under the GNU Lesser General Public License version 3 and version 2.1.

QtSpim

FP Regs Int Regs [10] Data Text

PC = 4194364 EPC = 4194368 Cause = 36 BadVAddr = 0 Status = 805371664

HI = 0 LO = 0

R0 [r0] = 0 R1 [at] = 268500992 R2 [v0] = 4 R3 [v1] = 0 R4 [a0] = 1 R5 [a1] = 2147483092 R6 [a2] = 2147483100 R7 [a3] = 0 R8 [t0] = -35 R9 [t1] = 18 R10 [t2] = 0 R11 [t3] = 0 R12 [t4] = 0 R13 [t5] = 0 R14 [t6] = 0 R15 [t7] = 0 R16 [s0] = 0 R17 [s1] = 0 R18 [s2] = 0 R19 [s3] = 0 R20 [s4] = 0 R21 [s5] = 0 R22 [s6] = 0 R23 [s7] = 0 R24 [t8] = 0 R25 [t9] = 0 R26 [k0] = 0 R27 [k1] = 0 R28 [gp] = 268468224 R29 [sp] = 2147483088 R30 [s8] = 0 R31 [ra] = 4194328

[0040000001] 0f640000 lw \$4, 0(\$29) ; 183: lw \$v0 0(\$sp) # argc
[00400004] 27650004 addiu \$5, \$29, 4 ; 184: addiu \$v1, \$sp 4 # argv
[00400004] 24e60004 addiu \$6, \$5, 4 ; 185: addiu \$v2, \$sp 4 # envp
[0040000c] 00041080 sll \$2, \$4, 2 ; 186: sll \$v0 \$v2 2
[00400010] 00c23021 addu \$6, \$6, \$2 ; 187: addu \$s2 \$s2 \$v0
[00400014] 0c100009 jal 0x00400024 [main] ; 188: jal main
[00400018] 00000000 nop ; 189: nop
[0040001c] 34020000 ori \$2, \$0, 10 ; 191: li \$v0 10
[00400020] 00000000 syscall ; 192: syscall # syscall 10 (exit)
[00400024] 3c011001 lui \$1, 4097 ; 193: lw \$t0, num1 # temp = num1
[00400028] x8c28000 lw \$8, 0(\$1) ; 194: lw \$t1, \$0, \$t0 # accum = temp
[00400030] 3c011000 addu \$9, \$8, 0 ; 195: add \$t1, \$0, \$t0 # accum = temp
[00400031] 3c011000 lui \$1, 4097 ; 196: lw \$t0, num2 # temp = num2
[00400033] x8c2800 lw \$8, 0(\$1) ; 197: addu \$s2 \$s2 \$v0
[00400035] x8c2800 lw \$8, 0(\$1) ; 198: addu \$s2 \$s2 \$v0
[00400037] x0128 Execution stopped at breakpoint at 0x00400037

Breakpoint Continue Single Step Abort

User Text Segment [00400000]..[00440000]
; 183: lw \$v0 0(\$sp) # argc
; 184: addiu \$v1, \$sp 4 # argv
; 185: addiu \$v2, \$sp 4 # envp
; 186: sll \$v0 \$v2 2
; 187: addu \$s2 \$s2 \$v0
; 188: jal main
; 189: nop
; 190: li \$v0 10
; 191: syscall # syscall 10 (exit)
; 192: lw \$t0, num1 # temp = num1
; 193: lui \$1, 4097
; 194: lw \$t1, \$0, \$t0 # accum = temp
; 195: add \$t1, \$0, \$t0 # accum = temp
; 196: lw \$t0, num2 # temp = num2
; 197: addu \$s2 \$s2 \$v0
; 198: addu \$s2 \$s2 \$v0
; 199: li \$v0 4 # syscall 4 (print_int)
; 200: syscall
; 201: li \$v0 1 # syscall 1 (print_int)
; 202: syscall
; 203: srl \$s0 \$k0 1 # Extract ExCode Field
; 204: andi \$s0 \$s0 0x1
; 205: li \$v0 1 # syscall 1 (print_int)
; 206: srl \$s0 \$k0 2 # Extract ExCode Field
; 207: andi \$s0 \$s0 0x1f
; 208: syscall
; 209: li \$v0 4 # syscall 4 (print_str)
; 210: syscall
; 211: andi \$s0 \$k0 0x3c
; 212: lw \$s0 _excp(\$s0)
; 213: li \$v0 4 # syscall 4 (print_str)
; 214: syscall
; 215: hlt

Kernel Text Segment [80000000]..[80010000]
; 90: move \$k1 \$at # Save \$at
; 92: sw \$v0 \$1 # Not re-entrant and we can't trust \$sp
; 93: ss \$s0 \$s2 # But we need to use these registers
; 94: mflo \$k0 \$13 # Cause register
; 96: srl \$s1 \$s0 \$k0 1 # Extract ExCode Field
; 97: andi \$s1 \$s0 0x1
; 98: andi \$s0 \$s0 0x1
; 101: li \$v0 4 # syscall 4 (print_int)
; 102: la \$s0 _m1
; 103: syscall
; 105: li \$v0 1 # syscall 1 (print_int)
; 106: srl \$s0 \$k0 2 # Extract ExCode Field
; 107: andi \$s0 \$s0 0x1f
; 108: syscall
; 109: li \$v0 4 # syscall 4 (print_str)
; 110: syscall
; 111: andi \$s0 \$k0 0x3c
; 112: lw \$s0 _excp(\$s0)
; 113: li \$v0 4 # syscall 4 (print_str)
; 114: syscall
; 115: hlt

All Rights Reserved.
SPIM is distributed under a BSD license.
See the file README for a full copyright notice.
QtSPIM is linked to the Qt library, which is distributed under the GNU Lesser General Public License version 3 and version 2.1.

QtSpim

FP Regs Console

PC = EPC = Cause = BadVAddr = Status =

HI = LO =

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

[00000011] 01c40000 lw \$2, 304(\$1)
[000001d0] 00000000 nop ; 113: nop
[000001d4] 0000000c syscall ; 114: syscall
[000001d9] 3a010010 ori \$1, \$0, 74 ; 115: hlt

All Rights Reserved.
SPIM is distributed under a BSD license.
See the file README for a full copyright notice.
QtSPIM is linked to the Qt library, which is distributed under the GNU Lesser General Public License version 3 and version 2.1.

2. Change of registers in sum2 .asm

QtSpim

Int Regs [10]

PC = 4194348	4194344	[00400000] 8fa40000 lw \$4, 0(\$29)	User Text Segment [00400000]..[00440000]
Cause = 36		[00400004] 27a50004 addiu \$5, \$29, 4	; 183: lw \$a0 0(\$sp) # argc
BadVAddr = 0		[00400008] 24a60004 addiu \$6, \$5, 4	; 184: addiu \$a1 \$sp 4 # argv
Status = 805371664		[0040000c] 00041080 lui \$2, \$4, 2	; 185: addiu \$a2 \$a1 4 # envp
HI = 0		[00400010] 00c23021 addu \$6, \$6, \$2	; 186: sll \$v0 \$a0 2
LO = 0		[00400014] 0c100009 jal 0x00400024 [main]	; 187: addu \$a2 \$a2 \$v0
R0 [r0] = 0		[00400018] 00000000 nop	; 188: jal main
R1 [at] = 268500992		[00400020] 3c011001 lui \$1, 4097	; 189: nop
R2 [v0] = 4		[00400024] x8c28000 lw \$8, 0(\$1)	; 190: syscall # syscall 10 (exit)
R3 [v1] = 0		[00400028] 1x0008482 add \$9, \$0, \$8	; 191: lw \$t0, num1 # temp = num1
R4 [a0] = 1		[00400030] 3c011001 lui \$1, 4097	; 192: lw \$t0, num2 # temp = num2
R5 [a1] = 2147483100		[00400031] Breakpoint	\$t1, \$t1, \$t0 # accum = accum + temp
R6 [a2] = 2147483108		[00400032] ; num3 # temp = num3	, num3 # temp = num3
R7 [t0] = 13		[00400034] ; Execution stopped at breakpoint at 0x0040	\$t1, \$t1, \$t0 # accum = accum + temp
R8 [t1] = 0		[00400036] ; , sum # sum = accum	, sum # sum = accum
R9 [t2] = 0		[00400038] Continue	v0, \$0, 1 # \$v0 = code for 'print-int'
R10 [t3] = 0		[00400040] Abort	0, \$0, \$t1 # \$a0 = accum
R11 [t4] = 0		[00400042] l # syscall(\$v0=1) prints \$a0	l # syscall(\$v0=1) prints \$a0
R12 [t5] = 0		[00400044] ; , \$t0 # return control to the simulator	, \$t0 # return control to the simulator
R13 [t6] = 0		[00400046] ; 20: jr \$v0 #	, 20: jr \$v0 # return control to the simulator
R14 [t7] = 0		[00400048] ; 21: addu \$t1, \$t0 # accum = accum + temp	, 21: addu \$t1, \$t0 # accum = accum + temp
R15 [t8] = 0		[0040004a] ; , sum # sum = accum	, sum # sum = accum
R16 [s0] = 0		[0040004c] ; 22: sw \$a0 \$2 # But we need to use these registers	Kernel Text Segment [80000000]..[80010000]
R17 [s1] = 0		[0040004e] 0001d821 addu \$27, \$0, \$1	; 90: move \$k1 \$at # Save \$at
R18 [s2] = 0		[00400050] 00019000 lui \$1, -28672	; 92: sw \$v0 \$1 # Not re-entrant and we can't trust \$sp
R19 [s3] = 0		[00400052] ac220000 sw \$2, \$12(\$1)	; 93: sw \$a0 \$2 # But we need to use these registers
R20 [s4] = 0		[00400054] 3c019000 lui \$1, -28672	
R21 [s5] = 0		[00400056] ac240204	
R22 [s6] = 0		[00400058] 401a6800 mfc0 \$26, \$13	; 95: mfc0 \$k0 \$13 # Cause register
R23 [s7] = 0		[0040005a] 001a2082 srl \$4, \$26, 2	; 96: srl \$a0 \$k0 2 # Extract ExcCode Field
R24 [t8] = 0		[0040005c] 30840001 andi \$4, \$4, 31	; 97: andi \$a0 \$a0 0x1f
R25 [t9] = 0		[0040005e] 34020004 ori \$2, \$0, 4	; 101: li \$v0 4 # syscall 4 (print_int)
R26 [k0] = 0		[00400060] 3c049000 lui \$4, -28672	; 102: la \$a0 _m1_
R27 [k1] = 0		[00400062] 00000000 j \$t1	; 103: syscall
R28 [gp] = 268468224		[00400064] 34020001 ori \$2, \$0, 1	; 105: li \$v0 1 # syscall 1 (print_int)
R29 [sp] = 2147483096		[00400066] 001a2082 srl \$4, \$26, 2	; 106: srl \$a0 \$k0 2 # Extract ExcCode Field
R30 [s8] = 0		[00400068] 30840001 andi \$4, \$4, 31	; 107: andi \$a0 \$a0 0x1f
R31 [ra] = 4194328		[00400070] 00000000 syscall	; 108: syscall

All Rights Reserved.
 SPIM is distributed under a BSD license.
 See the file README for a full copyright notice.
 QtSPIM is linked to the Qt library, which is distributed under the GNU Lesser General Public License version 3 and version 2.1.

QtSpim

Int Regs [10]

PC = 4194352	4194348	[00400000] 8fa40000 lw \$4, 0(\$29)	User Text Segment [00400000]..[00440000]
Cause = 36		[00400004] 27a50004 addiu \$5, \$29, 4	; 183: lw \$a0 0(\$sp) # argc
BadVAddr = 0		[00400008] 24a60004 addiu \$6, \$5, 4	; 184: addiu \$a1 \$sp 4 # argv
Status = 805371664		[0040000c] 00041080 lui \$2, \$4, 2	; 185: addiu \$a2 \$a1 4 # envp
HI = 0		[00400010] 00c23021 addu \$6, \$6, \$2	; 186: sll \$v0 \$a0 2
LO = 0		[00400014] 0c100009 jal 0x00400024 [main]	; 187: addu \$a2 \$a2 \$v0
R0 [r0] = 0		[00400018] 00000000 nop	; 188: jal main
R1 [at] = 268500992		[00400020] 3c011001 lui \$1, 4097	; 189: nop
R2 [v0] = 4		[00400024] x8c28000 lw \$8, 0(\$1)	; 190: syscall # syscall 10 (exit)
R3 [v1] = 0		[00400028] 1x0008482 add \$9, \$0, \$8	; 191: lw \$t0, num1 # temp = num1
R4 [a0] = 1		[00400030] 3c011001 lui \$1, 4097	; 192: lw \$t0, num2 # temp = num2
R5 [a1] = 2147483100		[00400031] Breakpoint	\$t1, \$t1, \$t0 # accum = accum + temp
R6 [a2] = 2147483108		[00400032] ; num3 # temp = num3	, num3 # temp = num3
R7 [t0] = 13		[00400034] ; Execution stopped at breakpoint at 0x0040	\$t1, \$t1, \$t0 # accum = accum + temp
R8 [t1] = 0		[00400036] ; , sum # sum = accum	, sum # sum = accum
R9 [t2] = 0		[00400038] Continue	v0, \$0, 1 # \$v0 = code for 'print-int'
R10 [t3] = 0		[00400040] Abort	0, \$0, \$t1 # \$a0 = accum
R11 [t4] = 0		[00400042] l # syscall(\$v0=1) prints \$a0	l # syscall(\$v0=1) prints \$a0
R12 [t5] = 0		[00400044] ; , \$t0 # return control to the simulator	, \$t0 # return control to the simulator
R13 [t6] = 0		[00400046] ; 20: jr \$v0 #	, 20: jr \$v0 # return control to the simulator
R14 [t7] = 0		[00400048] ; 21: addu \$t1, \$t0 # accum = accum + temp	, 21: addu \$t1, \$t0 # accum = accum + temp
R15 [t8] = 0		[0040004a] ; , sum # sum = accum	, sum # sum = accum
R16 [s0] = 0		[0040004c] ; 22: sw \$a0 \$2 # But we need to use these registers	Kernel Text Segment [80000000]..[80010000]
R17 [s1] = 0		[0040004e] 0001d821 addu \$27, \$0, \$1	; 90: move \$k1 \$at # Save \$at
R18 [s2] = 0		[00400050] 00019000 lui \$1, -28672	; 92: sw \$v0 \$1 # Not re-entrant and we can't trust \$sp
R19 [s3] = 0		[00400052] ac220000 sw \$2, \$12(\$1)	; 93: sw \$a0 \$2 # But we need to use these registers
R20 [s4] = 0		[00400054] 3c019000 lui \$1, -28672	
R21 [s5] = 0		[00400056] ac240204	
R22 [s6] = 0		[00400058] 401a6800 mfc0 \$26, \$13	; 95: mfc0 \$k0 \$13 # Cause register
R23 [s7] = 0		[00400060] 001a2082 srl \$4, \$26, 2	; 96: srl \$a0 \$k0 2 # Extract ExcCode Field
R24 [t8] = 0		[00400062] 30840001 andi \$4, \$4, 31	; 97: andi \$a0 \$a0 0x1f
R25 [t9] = 0		[00400064] 34020004 ori \$2, \$0, 4	; 101: li \$v0 4 # syscall 4 (print_int)
R26 [k0] = 0		[00400066] 3c049000 lui \$4, -28672	; 102: la \$a0 _m1_
R27 [k1] = 0		[00400068] 00000000 j \$t1	; 103: syscall
R28 [gp] = 268468224		[00400070] 34020001 ori \$2, \$0, 1	; 105: li \$v0 1 # syscall 1 (print_int)
R29 [sp] = 2147483096		[00400072] 001a2082 srl \$4, \$26, 2	; 106: srl \$a0 \$k0 2 # Extract ExcCode Field
R30 [s8] = 0		[00400074] 30840001 andi \$4, \$4, 31	; 107: andi \$a0 \$a0 0x1f
R31 [ra] = 4194328		[00400076] 00000000 syscall	; 108: syscall

All Rights Reserved.
 SPIM is distributed under a BSD license.
 See the file README for a full copyright notice.
 QtSPIM is linked to the Qt library, which is distributed under the GNU Lesser General Public License version 3 and version 2.1.

QtSpim

FP Regs **Int Regs [10]**

Data **Text**

Int Regs [10]

PC = 4194372
EPC = 4194368
Cause = 36
BadVAddr = 0
Status = 805371664

HI = 0
LO = 0

R0 [r0] = 0
R1 [at] = 268500992
R2 [v0] = 4
R3 [v1] = 0
R4 [a0] = 1
R5 [a1] = 2147483100
R6 [a2] = 2147483108
R7 [a3] = 0
R8 [t0] = 272
R9 [t1] = -17
R10 [t2] = 0
R11 [t3] = 0
R12 [t4] = 0
R13 [t5] = 0
R14 [t6] = 0
R15 [t7] = 0
R16 [s0] = 0
R17 [s1] = 0
R18 [s2] = 0
R19 [s3] = 0
R20 [s4] = 0
R21 [s5] = 0
R22 [s6] = 0
R23 [s7] = 0
R24 [t8] = 0
R25 [t9] = 0
R26 [k0] = 0
R27 [k1] = 0
R28 [gp] = 268468224
R29 [sp] = 2147483096
R30 [s8] = 0
R31 [ra] = 4194328

User Text Segment [00400000]..[00440000]
 [00400000] 8fa00000 lw \$4, 0(\$29)
 [00400004] 27a50004 addiu \$5, \$29, 4
 [00400008] 24a60004 addiu \$6, \$5, 4
 [0040000c] 00041880 sll \$2, \$4, 2
 [00400010] 002c2021 addu \$6, \$6, \$2
 [00400014] 0c100009 jal 0x00400024 [main]
 [00400018] 00000000 nop
 [0040001c] 3402000a ori \$2, \$0, 10
 [00400020] 00000000 syscall
 [00400024] 00011001 lui \$1, 4097
 [00400028] x8c28000 lw \$8, 0(\$1)
 [0040002c] x00040002 add \$9, \$0, \$8 ; 19: add \$t1, \$0, \$t0 # accum = temp
 [00400030] 3c011801 lui \$1. 4097 ; 20: lw \$t6, num2 # temp = num2
 [00400034] 00000000 Breakpoint ; t1, \$t1, \$t0 # accum = accum + temp
 [00400038] 00000000 , num3 # temp = num3
 [00400040] Execution stopped at breakpoint at 0x0040 ; t1, \$t1, \$t0 # accum = accum + temp
 [00400044] , sum # sum = accum
 [00400048] 00000000 ib0, \$0, 1 # \$v0 = code for 'print-int'
 [00400050] 00000000 0, \$0, \$t1 # \$a0 = accum
 [00400054] l # syscall(\$v0=1) prints \$a0
 [00400058] 00000000 ; 20: li \$a1 # return control to the simulator
 [0040005c] 00000000 j \$32 ; 20: li \$a1 # return control to the simulator
 Kernel Text Segment [80000000]..[80010000]
 [00000180] 0001d821 addu \$27, \$0, \$1 ; 90: move \$k1 \$at # Save \$at
 [00000184] 3c019000 lui \$1, -28672 ; 92: sw \$v0 \$1 # Not re-entrant and we can't trust \$sp
 [00000188] ac220200 sw \$2, 512(\$1)
 [0000018c] 3c019000 lui \$1, -28672 ; 93: sw \$a0 \$2 # But we need to use these registers
 [00000190] ac240204 sw \$4, 516(\$1)
 [00000194] 401a6800 mfc0 \$26, \$13 ; 95: mfc0 \$k0 \$13 # Cause register
 [00000198] 001a2882 srl \$4, \$26, 2 ; 96: srl \$a0 \$k0 2 # Extract ExcCode Field
 [0000019c] 3084001f andi \$4, \$4, 31 ; 97: andi \$a0 \$a0 0x1f
 [000001a0] 34020004 ori \$2, \$0, 4 ; 101: li \$v0 4 # syscall 4 (print_str)
 [000001a4] 3c049000 lui \$4, -28672 [_m1_] ; 102: la \$a0 _m1_
 [000001a8] 00000000 syscall ; 103: syscall
 [000001bc] 34020001 ori \$2, \$0, 1 ; 105: li \$v0 1 # syscall 1 (print_int)
 [000001b0] 001a2082 srl \$4, \$26, 2 ; 106: sw \$a0 \$k0 2 # Extract ExcCode Field
 [000001b4] 3084001f andi \$4, \$4, 31 ; 107: andi \$a0 \$a0 0x3c
 [000001b8] 00000000 syscall ; 108: syscall
 [000001bc] 34020004 ori \$2, \$0, 4 ; 110: li \$v0 4 # syscall 4 (print_str)
 [000001c0] 3244003c andi \$4, \$26, 60 ; 111: andi \$a0 \$k0 0x3c
 [000001c4] 3c019000 lui \$1, -28672 ; 112: lw \$a0 __exc(\$a0)
 [000001c8] 00240021 addu \$1, \$1, \$4 ; 113: nop
 [000001cc] 8c240180 lw \$4, 384(\$1) ; 114: syscall
 [000001d0] 00000000 nop ; 115: bne \$a0 \$a1 \$a2 ; 116: bne \$a0 \$a1 \$a2 ; 117: bne \$a0 \$a1 \$a2 ; 118: bne \$a0 \$a1 \$a2 ; 119: bne \$a0 \$a1 \$a2 ; 120: bne \$a0 \$a1 \$a2 ; 121: bne \$a0 \$a1 \$a2 ; 122: lw \$a0 __exc(\$a0); 123: bne \$a0 \$a1 \$a2 ; 124: bne \$a0 \$a1 \$a2 ; 125: bne \$a0 \$a1 \$a2 ; 126: bne \$a0 \$a1 \$a2 ; 127: bne \$a0 \$a1 \$a2 ; 128: bne \$a0 \$a1 \$a2 ; 129: bne \$a0 \$a1 \$a2 ; 130: bne \$a0 \$a1 \$a2 ; 131: bne \$a0 \$a1 \$a2 ; 132: bne \$a0 \$a1 \$a2 ; 133: bne \$a0 \$a1 \$a2 ; 134: bne \$a0 \$a1 \$a2 ; 135: bne \$a0 \$a1 \$a2 ; 136: bne \$a0 \$a1 \$a2 ; 137: bne \$a0 \$a1 \$a2 ; 138: bne \$a0 \$a1 \$a2 ; 139: bne \$a0 \$a1 \$a2 ; 140: bne \$a0 \$a1 \$a2 ; 141: bne \$a0 \$a1 \$a2 ; 142: bne \$a0 \$a1 \$a2 ; 143: bne \$a0 \$a1 \$a2 ; 144: bne \$a0 \$a1 \$a2 ; 145: bne \$a0 \$a1 \$a2 ; 146: bne \$a0 \$a1 \$a2 ; 147: bne \$a0 \$a1 \$a2 ; 148: bne \$a0 \$a1 \$a2 ; 149: bne \$a0 \$a1 \$a2 ; 150: bne \$a0 \$a1 \$a2 ; 151: bne \$a0 \$a1 \$a2 ; 152: bne \$a0 \$a1 \$a2 ; 153: bne \$a0 \$a1 \$a2 ; 154: bne \$a0 \$a1 \$a2 ; 155: bne \$a0 \$a1 \$a2 ; 156: bne \$a0 \$a1 \$a2 ; 157: bne \$a0 \$a1 \$a2 ; 158: bne \$a0 \$a1 \$a2 ; 159: bne \$a0 \$a1 \$a2 ; 160: bne \$a0 \$a1 \$a2 ; 161: bne \$a0 \$a1 \$a2 ; 162: bne \$a0 \$a1 \$a2 ; 163: bne \$a0 \$a1 \$a2 ; 164: bne \$a0 \$a1 \$a2 ; 165: bne \$a0 \$a1 \$a2 ; 166: bne \$a0 \$a1 \$a2 ; 167: bne \$a0 \$a1 \$a2 ; 168: bne \$a0 \$a1 \$a2 ; 169: bne \$a0 \$a1 \$a2 ; 170: bne \$a0 \$a1 \$a2 ; 171: bne \$a0 \$a1 \$a2 ; 172: bne \$a0 \$a1 \$a2 ; 173: bne \$a0 \$a1 \$a2 ; 174: bne \$a0 \$a1 \$a2 ; 175: bne \$a0 \$a1 \$a2 ; 176: bne \$a0 \$a1 \$a2 ; 177: bne \$a0 \$a1 \$a2 ; 178: bne \$a0 \$a1 \$a2 ; 179: bne \$a0 \$a1 \$a2 ; 180: bne \$a0 \$a1 \$a2 ; 181: bne \$a0 \$a1 \$a2 ; 182: bne \$a0 \$a1 \$a2 ; 183: bne \$a0 \$a1 \$a2 ; 184: bne \$a0 \$a1 \$a2 ; 185: bne \$a0 \$a1 \$a2 ; 186: bne \$a0 \$a1 \$a2 ; 187: bne \$a0 \$a1 \$a2 ; 188: bne \$a0 \$a1 \$a2 ; 189: bne \$a0 \$a1 \$a2 ; 190: bne \$a0 \$a1 \$a2 ; 191: bne \$a0 \$a1 \$a2 ; 192: bne \$a0 \$a1 \$a2 ; 193: bne \$a0 \$a1 \$a2 ; 194: bne \$a0 \$a1 \$a2 ; 195: bne \$a0 \$a1 \$a2 ; 196: bne \$a0 \$a1 \$a2 ; 197: bne \$a0 \$a1 \$a2 ; 198: bne \$a0 \$a1 \$a2 ; 199: bne \$a0 \$a1 \$a2 ; 200: bne \$a0 \$a1 \$a2 ; 201: bne \$a0 \$a1 \$a2 ; 202: bne \$a0 \$a1 \$a2 ; 203: bne \$a0 \$a1 \$a2 ; 204: bne \$a0 \$a1 \$a2 ; 205: bne \$a0 \$a1 \$a2 ; 206: bne \$a0 \$a1 \$a2 ; 207: bne \$a0 \$a1 \$a2 ; 208: bne \$a0 \$a1 \$a2 ; 209: bne \$a0 \$a1 \$a2 ; 210: bne \$a0 \$a1 \$a2 ; 211: bne \$a0 \$a1 \$a2 ; 212: bne \$a0 \$a1 \$a2 ; 213: bne \$a0 \$a1 \$a2 ; 214: bne \$a0 \$a1 \$a2 ; 215: bne \$a0 \$a1 \$a2 ; 216: bne \$a0 \$a1 \$a2 ; 217: bne \$a0 \$a1 \$a2 ; 218: bne \$a0 \$a1 \$a2 ; 219: bne \$a0 \$a1 \$a2 ; 220: bne \$a0 \$a1 \$a2 ; 221: bne \$a0 \$a1 \$a2 ; 222: bne \$a0 \$a1 \$a2 ; 223: bne \$a0 \$a1 \$a2 ; 224: bne \$a0 \$a1 \$a2 ; 225: bne \$a0 \$a1 \$a2 ; 226: bne \$a0 \$a1 \$a2 ; 227: bne \$a0 \$a1 \$a2 ; 228: bne \$a0 \$a1 \$a2 ; 229: bne \$a0 \$a1 \$a2 ; 230: bne \$a0 \$a1 \$a2 ; 231: bne \$a0 \$a1 \$a2 ; 232: bne \$a0 \$a1 \$a2 ; 233: bne \$a0 \$a1 \$a2 ; 234: bne \$a0 \$a1 \$a2 ; 235: bne \$a0 \$a1 \$a2 ; 236: bne \$a0 \$a1 \$a2 ; 237: bne \$a0 \$a1 \$a2 ; 238: bne \$a0 \$a1 \$a2 ; 239: bne \$a0 \$a1 \$a2 ; 240: bne \$a0 \$a1 \$a2 ; 241: bne \$a0 \$a1 \$a2 ; 242: bne \$a0 \$a1 \$a2 ; 243: bne \$a0 \$a1 \$a2 ; 244: bne \$a0 \$a1 \$a2 ; 245: bne \$a0 \$a1 \$a2 ; 246: bne \$a0 \$a1 \$a2 ; 247: bne \$a0 \$a1 \$a2 ; 248: bne \$a0 \$a1 \$a2 ; 249: bne \$a0 \$a1 \$a2 ; 250: bne \$a0 \$a1 \$a2 ; 251: bne \$a0 \$a1 \$a2 ; 252: bne \$a0 \$a1 \$a2 ; 253: bne \$a0 \$a1 \$a2 ; 254: bne \$a0 \$a1 \$a2 ; 255: bne \$a0 \$a1 \$a2 ; 256: bne \$a0 \$a1 \$a2 ; 257: bne \$a0 \$a1 \$a2 ; 258: bne \$a0 \$a1 \$a2 ; 259: bne \$a0 \$a1 \$a2 ; 260: bne \$a0 \$a1 \$a2 ; 261: bne \$a0 \$a1 \$a2 ; 262: bne \$a0 \$a1 \$a2 ; 263: bne \$a0 \$a1 \$a2 ; 264: bne \$a0 \$a1 \$a2 ; 265: bne \$a0 \$a1 \$a2 ; 266: bne \$a0 \$a1 \$a2 ; 267: bne \$a0 \$a1 \$a2 ; 268: bne \$a0 \$a1 \$a2 ; 269: bne \$a0 \$a1 \$a2 ; 270: bne \$a0 \$a1 \$a2 ; 271: bne \$a0 \$a1 \$a2 ; 272: bne \$a0 \$a1 \$a2 ; 273: bne \$a0 \$a1 \$a2 ; 274: bne \$a0 \$a1 \$a2 ; 275: bne \$a0 \$a1 \$a2 ; 276: bne \$a0 \$a1 \$a2 ; 277: bne \$a0 \$a1 \$a2 ; 278: bne \$a0 \$a1 \$a2 ; 279: bne \$a0 \$a1 \$a2 ; 280: bne \$a0 \$a1 \$a2 ; 281: bne \$a0 \$a1 \$a2 ; 282: bne \$a0 \$a1 \$a2 ; 283: bne \$a0 \$a1 \$a2 ; 284: bne \$a0 \$a1 \$a2 ; 285: bne \$a0 \$a1 \$a2 ; 286: bne \$a0 \$a1 \$a2 ; 287: bne \$a0 \$a1 \$a2 ; 288: bne \$a0 \$a1 \$a2 ; 289: bne \$a0 \$a1 \$a2 ; 290: bne \$a0 \$a1 \$a2 ; 291: bne \$a0 \$a1 \$a2 ; 292: bne \$a0 \$a1 \$a2 ; 293: bne \$a0 \$a1 \$a2 ; 294: bne \$a0 \$a1 \$a2 ; 295: bne \$a0 \$a1 \$a2 ; 296: bne \$a0 \$a1 \$a2 ; 297: bne \$a0 \$a1 \$a2 ; 298: bne \$a0 \$a1 \$a2 ; 299: bne \$a0 \$a1 \$a2 ; 300: bne \$a0 \$a1 \$a2 ; 301: bne \$a0 \$a1 \$a2 ; 302: bne \$a0 \$a1 \$a2 ; 303: bne \$a0 \$a1 \$a2 ; 304: bne \$a0 \$a1 \$a2 ; 305: bne \$a0 \$a1 \$a2 ; 306: bne \$a0 \$a1 \$a2 ; 307: bne \$a0 \$a1 \$a2 ; 308: bne \$a0 \$a1 \$a2 ; 309: bne \$a0 \$a1 \$a2 ; 310: bne \$a0 \$a1 \$a2 ; 311: bne \$a0 \$a1 \$a2 ; 312: bne \$a0 \$a1 \$a2 ; 313: bne \$a0 \$a1 \$a2 ; 314: bne \$a0 \$a1 \$a2 ; 315: bne \$a0 \$a1 \$a2 ; 316: bne \$a0 \$a1 \$a2 ; 317: bne \$a0 \$a1 \$a2 ; 318: bne \$a0 \$a1 \$a2 ; 319: bne \$a0 \$a1 \$a2 ; 320: bne \$a0 \$a1 \$a2 ; 321: bne \$a0 \$a1 \$a2 ; 322: bne \$a0 \$a1 \$a2 ; 323: bne \$a0 \$a1 \$a2 ; 324: bne \$a0 \$a1 \$a2 ; 325: bne \$a0 \$a1 \$a2 ; 326: bne \$a0 \$a1 \$a2 ; 327: bne \$a0 \$a1 \$a2 ; 328: bne \$a0 \$a1 \$a2 ; 329: bne \$a0 \$a1 \$a2 ; 330: bne \$a0 \$a1 \$a2 ; 331: bne \$a0 \$a1 \$a2 ; 332: bne \$a0 \$a1 \$a2 ; 333: bne \$a0 \$a1 \$a2 ; 334: bne \$a0 \$a1 \$a2 ; 335: bne \$a0 \$a1 \$a2 ; 336: bne \$a0 \$a1 \$a2 ; 337: bne \$a0 \$a1 \$a2 ; 338: bne \$a0 \$a1 \$a2 ; 339: bne \$a0 \$a1 \$a2 ; 340: bne \$a0 \$a1 \$a2 ; 341: bne \$a0 \$a1 \$a2 ; 342: bne \$a0 \$a1 \$a2 ; 343: bne \$a0 \$a1 \$a2 ; 344: bne \$a0 \$a1 \$a2 ; 345: bne \$a0 \$a1 \$a2 ; 346: bne \$a0 \$a1 \$a2 ; 347: bne \$a0 \$a1 \$a2 ; 348: bne \$a0 \$a1 \$a2 ; 349: bne \$a0 \$a1 \$a2 ; 350: bne \$a0 \$a1 \$a2 ; 351: bne \$a0 \$a1 \$a2 ; 352: bne \$a0 \$a1 \$a2 ; 353: bne \$a0 \$a1 \$a2 ; 354: bne \$a0 \$a1 \$a2 ; 355: bne \$a0 \$a1 \$a2 ; 356: bne \$a0 \$a1 \$a2 ; 357: bne \$a0 \$a1 \$a2 ; 358: bne \$a0 \$a1 \$a2 ; 359: bne \$a0 \$a1 \$a2 ; 360: bne \$a0 \$a1 \$a2 ; 361: bne \$a0 \$a1 \$a2 ; 362: bne \$a0 \$a1 \$a2 ; 363: bne \$a0 \$a1 \$a2 ; 364: bne \$a0 \$a1 \$a2 ; 365: bne \$a0 \$a1 \$a2 ; 366: bne \$a0 \$a1 \$a2 ; 367: bne \$a0 \$a1 \$a2 ; 368: bne \$a0 \$a1 \$a2 ; 369: bne \$a0 \$a1 \$a2 ; 370: bne \$a0 \$a1 \$a2 ; 371: bne \$a0 \$a1 \$a2 ; 372: bne \$a0 \$a1 \$a2 ; 373: bne \$a0 \$a1 \$a2 ; 374: bne \$a0 \$a1 \$a2 ; 375: bne \$a0 \$a1 \$a2 ; 376: bne \$a0 \$a1 \$a2 ; 377: bne \$a0 \$a1 \$a2 ; 378: bne \$a0 \$a1 \$a2 ; 379: bne \$a0 \$a1 \$a2 ; 380: bne \$a0 \$a1 \$a2 ; 381: bne \$a0 \$a1 \$a2 ; 382: bne \$a0 \$a1 \$a2 ; 383: bne \$a0 \$a1 \$a2 ; 384: bne \$a0 \$a1 \$a2 ; 385: bne \$a0 \$a1 \$a2 ; 386: bne \$a0 \$a1 \$a2 ; 387: bne \$a0 \$a1 \$a2 ; 388: bne \$a0 \$a1 \$a2 ; 389: bne \$a0 \$a1 \$a2 ; 390: bne \$a0 \$a1 \$a2 ; 391: bne \$a0 \$a1 \$a2 ; 392: bne \$a0 \$a1 \$a2 ; 393: bne \$a0 \$a1 \$a2 ; 394: bne \$a0 \$a1 \$a2 ; 395: bne \$a0 \$a1 \$a2 ; 396: bne \$a0 \$a1 \$a2 ; 397: bne \$a0 \$a1 \$a2 ; 398: bne \$a0 \$a1 \$a2 ; 399: bne \$a0 \$a1 \$a2 ; 400: bne \$a0 \$a1 \$a2 ; 401: bne \$a0 \$a1 \$a2 ; 402: bne \$a0 \$a1 \$a2 ; 403: bne \$a0 \$a1 \$a2 ; 404: bne \$a0 \$a1 \$a2 ; 405: bne \$a0 \$a1 \$a2 ; 406: bne \$a0 \$a1 \$a2 ; 407: bne \$a0 \$a1 \$a2 ; 408: bne \$a0 \$a1 \$a2 ; 409: bne \$a0 \$a1 \$a2 ; 410: bne \$a0 \$a1 \$a2 ; 411: bne \$a0 \$a1 \$a2 ; 412: bne \$a0 \$a1 \$a2 ; 413: bne \$a0 \$a1 \$a2 ; 414: bne \$a0 \$a1 \$a2 ; 415: bne \$a0 \$a1 \$a2 ; 416: bne \$a0 \$a1 \$a2 ; 417: bne \$a0 \$a1 \$a2 ; 418: bne \$a0 \$a1 \$a2 ; 419: bne \$a0 \$a1 \$a2 ; 420: bne \$a0 \$a1 \$a2 ; 421: bne \$a0 \$a1 \$a2 ; 422: bne \$a0 \$a1 \$a2 ; 423: bne \$a0 \$a1 \$a2 ; 424: bne \$a0 \$a1 \$a2 ; 425: bne \$a0 \$a1 \$a2 ; 426: bne \$a0 \$a1 \$a2 ; 427: bne \$a0 \$a1 \$a2 ; 428: bne \$a0 \$a1 \$a2 ; 429: bne \$a0 \$a1 \$a2 ; 430: bne \$a0 \$a1 \$a2 ; 431: bne \$a0 \$a1 \$a2 ; 432: bne \$a0 \$a1 \$a2 ; 433: bne \$a0 \$a1 \$a2 ; 434: bne \$a0 \$a1 \$a2 ; 435: bne \$a0 \$a1 \$a2 ; 436: bne \$a0 \$a1 \$a2 ; 437: bne \$a0 \$a1 \$a2 ; 438: bne \$a0 \$a1 \$a2 ; 439: bne \$a0 \$a1 \$a2 ; 440: bne \$a0 \$a1 \$a2 ; 441: bne \$a0 \$a1 \$a2 ; 442: bne \$a0 \$a1 \$a2 ; 443: bne \$a0 \$a1 \$a2 ; 444: bne \$a0 \$a1 \$a2 ; 445: bne \$a0 \$a1 \$a2 ; 446: bne \$a0 \$a1 \$a2 ; 447: bne \$a0 \$a1 \$a2 ; 448: bne \$a0 \$a1 \$a2 ; 449: bne \$a0 \$a1 \$a2 ; 450: bne \$a0 \$a1 \$a2 ; 451: bne \$a0 \$a1 \$a2 ; 452: bne \$a0 \$a1 \$a2 ; 453: bne \$a0 \$a1 \$a2 ; 454: bne \$a0 \$a1 \$a2 ; 455: bne \$a0 \$a1 \$a2 ; 456: bne \$a0 \$a1 \$a2 ; 457: bne \$a0 \$a1 \$a2 ; 458: bne \$a0 \$a1 \$a2 ; 459: bne \$a0 \$a1 \$a2 ; 460: bne \$a0 \$a1 \$a2 ; 461: bne \$a0 \$a1 \$a2 ; 462: bne \$a0 \$a1 \$a2 ; 463: bne \$a0 \$a1 \$a2 ; 464: bne \$a0 \$a1 \$a2 ; 465: bne \$a0 \$a1 \$a2 ; 466: bne \$a0 \$a1 \$a2 ; 467: bne \$a0 \$a1 \$a2 ; 468: bne \$a0 \$a1 \$a2 ; 469: bne \$a0 \$a1 \$a2 ; 470: bne \$a0 \$a1 \$a2 ; 471: bne \$a0 \$a1 \$a2 ; 472: bne \$a0 \$a1 \$a2 ; 473: bne \$a0 \$a1 \$a2 ; 474: bne \$a0 \$a1 \$a2 ; 475: bne \$a0 \$a1 \$a2 ; 476: bne \$a0 \$a1 \$a2 ; 477: bne \$a0 \$a1 \$a2 ; 478: bne \$a0 \$a1 \$a2 ; 479: bne \$a0 \$a1 \$a2 ; 480: bne \$a0 \$a1 \$a2 ; 481: bne \$a0 \$a1 \$a2 ; 482: bne \$a0 \$a1 \$a2 ; 483: bne \$a0 \$a1 \$a2 ; 484: bne \$a0 \$a1 \$a2 ; 485: bne \$a0 \$a1 \$a2 ; 486: bne \$a0 \$a1 \$a2 ; 487: bne \$a0 \$a1 \$a2 ; 488: bne \$a0 \$a1 \$a2 ; 489: bne \$a0 \$a1 \$a2 ; 490: bne \$a0 \$a1 \$a2 ; 491: bne \$a0 \$a1 \$a2 ; 492: bne \$a0 \$a1 \$a2 ; 493: bne \$a0 \$a1 \$a2 ; 494: bne \$a0 \$a1 \$a2 ; 495: bne \$a0 \$a1 \$a2 ; 496: bne \$a0 \$a1 \$a2 ; 497: bne \$a0 \$a1 \$a2 ; 498: bne \$a0 \$a1 \$a2 ; 499: bne \$a0 \$a1 \$a2 ; 500: bne \$a0 \$a1 \$a2 ; 501: bne \$a0 \$a1 \$a2 ; 502: bne \$a0 \$a1 \$a2 ; 503: bne \$a0 \$a1 \$a2 ; 504: bne \$a0 \$a1 \$a2 ; 505: bne \$a0 \$a1 \$a2 ; 506: bne \$a0 \$a1 \$a2 ; 507: bne \$a0 \$a1 \$a2 ; 508: bne \$a0 \$a1 \$a2 ; 509: bne \$a0 \$a1 \$a2 ; 510: bne \$a0 \$a1 \$a2 ; 511: bne \$a0 \$a1 \$a2 ; 512: bne \$a0 \$a1 \$a2 ; 513: bne \$a0 \$a1 \$a2 ; 514: bne \$a0 \$a1 \$a2 ; 515: bne \$a0 \$a1 \$a2 ; 516: bne \$a0 \$a1 \$a2 ; 517: bne \$a0 \$a1 \$a2 ; 518: bne \$a0 \$a1 \$a2 ; 519: bne \$a0 \$a1 \$a2 ; 520: bne \$a0 \$a1 \$a2 ; 521: bne \$a0 \$a1 \$a2 ; 522: bne \$a0 \$a1 \$a2 ; 523: bne \$a0 \$a1 \$a2 ; 524: bne \$a0 \$a1 \$a2 ; 525: bne \$a0 \$a1 \$a2 ; 526: bne \$a0 \$a1 \$a2 ; 527: bne \$a0 \$a1 \$a2 ; 528: bne \$a0 \$a1 \$a2 ; 529: bne \$a0 \$a1 \$a2 ; 530: bne \$a0 \$a1 \$a2 ; 531: bne \$a0 \$a1 \$a2 ; 532: bne \$a0 \$a1 \$a2 ; 533: bne \$a0 \$a1 \$a2 ; 534: bne \$a0 \$a1 \$a2 ; 535: bne \$a0 \$a1 \$a2 ; 536: bne \$a0 \$a1 \$a2 ; 537: bne \$a0 \$a1 \$a2 ; 538: bne \$a0 \$a1 \$a2 ; 539: bne \$a0 \$a1 \$a2 ; 540: bne \$a0 \$a1 \$a2 ; 541: bne \$a0 \$a1 \$a2 ; 542: bne \$a0 \$a1 \$a2 ; 543: bne \$a0 \$a1 \$a2 ; 544: bne \$a0 \$a1 \$a2 ; 545: bne \$a0 \$a1 \$a2 ; 546: bne \$a0 \$a1 \$a2 ; 547: bne \$a0 \$a1 \$a2 ; 548: bne \$a0 \$a1 \$a2 ; 549: bne \$a0 \$a1 \$a2 ; 550: bne \$a0 \$a1 \$a2 ; 551: bne \$a0 \$a1 \$a2 ; 552: bne \$a0 \$a1 \$a2 ; 553: bne \$a0 \$a1 \$a2 ; 554: bne \$a0 \$a1 \$a2 ; 555: bne \$a0 \$a1 \$a2 ; 556: bne \$a0 \$a1 \$a2 ; 557: bne \$a0 \$a1 \$a2 ; 558: bne \$a0 \$a1 \$a2 ; 559: bne \$a0 \$a1 \$a2 ; 560: bne \$a0 \$a1 \$a2 ; 561: bne \$a0 \$a1 \$a2 ; 562: bne \$a0 \$a1 \$a2 ; 563: bne \$a0 \$a1 \$a2 ; 564: bne \$a0 \$a1 \$a2 ; 565: bne \$a0 \$a1 \$a2 ; 566: bne \$a0 \$a1 \$a2 ; 567: bne \$a0 \$a1 \$a2 ; 568: bne \$a0 \$a1 \$a2 ; 569: bne \$a0 \$a1 \$a2 ; 570: bne \$a0 \$a1 \$a2 ; 571: bne \$a0 \$a1 \$a2 ; 572: bne \$a0 \$a1 \$a2 ; 573: bne \$a0 \$a1 \$a2 ; 574: bne \$a0 \$a1 \$a2 ; 575: bne \$a0 \$a1 \$a2 ; 576: bne \$a0 \$a1 \$a2 ; 577: bne \$a0 \$a1 \$a2 ; 578: bne \$a0 \$a1 \$a2 ; 579: bne \$a0 \$a1 \$a2 ; 580: bne \$a0 \$a1 \$a2 ; 581: bne \$a0 \$a1 \$a2 ; 582: bne \$a0 \$a1 \$a2 ; 583: bne \$a0 \$a1 \$a2 ; 584: bne \$a0 \$a1 \$a2 ; 585: bne \$a0 \$a1 \$a2 ; 586: bne \$a0 \$a1 \$a2 ; 587: bne \$a0 \$a1 \$a2 ; 588: bne \$a0 \$a1 \$a2 ; 589: bne \$a0 \$a1 \$a2 ; 590: bne \$a0 \$a1 \$a2 ; 591: bne \$a0 \$a1 \$a2 ; 592: bne \$a0 \$a1 \$a2 ; 593: bne \$a0 \$a1 \$a2 ; 594: bne \$a0 \$a1 \$a2 ; 595: bne \$a0 \$a1 \$a2 ; 596: bne \$a0 \$a1 \$a2 ; 597: bne \$a0 \$a1 \$a2 ; 598: bne \$a0 \$a1 \$a2 ; 599: bne \$a0 \$a1 \$a2 ; 600: bne \$a0 \$a1 \$a2 ; 601: bne \$a0 \$a1 \$a2 ; 602: bne \$a0 \$a1 \$a2 ; 603: bne \$a0 \$a1 \$a2 ; 604: bne \$a0 \$a1 \$a2 ; 605: bne \$a0 \$a1 \$a2 ; 606: bne \$a0 \$a1 \$a2 ; 607: bne \$a0 \$a1 \$a2 ; 608: bne \$a0 \$a1 \$a2 ; 609: bne \$a0 \$a1 \$a2 ; 610: bne \$a0 \$a1 \$a2 ; 611: bne \$a0 \$a1 \$a2 ; 612: bne \$a0 \$a1 \$a2 ; 613: bne \$a0 \$a1 \$a2 ; 614: bne \$a0 \$a1 \$a2 ; 615: bne \$a0 \$a1 \$a2 ; 616: bne \$a0 \$a1 \$a2 ; 617: bne \$a0 \$a1 \$a2 ; 618: bne \$a0 \$a1 \$a2 ; 619: bne \$a0 \$a1 \$a2 ; 620: bne \$a0 \$a1 \$a2 ; 621: bne \$a0 \$a1 \$a2 ; 622: bne \$a0 \$a1 \$a2 ; 623: bne \$a0 \$a1 \$a2 ; 624: bne \$a0 \$a1 \$a2 ; 625: bne \$a0 \$a1 \$a2 ; 626: bne \$a0 \$a1 \$a2 ; 627: bne \$a0 \$a1 \$a2 ; 628: bne \$a0 \$a1 \$a2 ; 629: bne \$a0 \$a1 \$a2 ; 630: bne \$a0 \$a1 \$a2 ; 631: bne \$a0 \$a1 \$a2 ; 632: bne \$a0 \$a1 \$a2 ; 633: bne \$a0 \$a1 \$a2 ; 634: bne \$a0 \$a1 \$a2 ; 635: bne \$a0 \$a1 \$a2 ; 636: bne

3. Change of registers in swap .asm

FP Regs Int Regs [10] Data Text

User Stack [7ffffd78]..[80000000]

```

PC = 4194336
EPC = 0
Cause = 0
BadVAddr = 0
Status = 805371664
HI = 0
LO = 0

R0 [r0] = 0
R1 [at] = 269580992
R2 [v0] = 10
R3 [v1] = 0
R4 [a0] = 1
R5 [a1] = 2147483004
R6 [a2] = 2147483012
R7 [a3] = 0
R8 [t0] = 269580992
R9 [t1] = 269580996
R10 [t2] = 269581008
R11 [t3] = 0
R12 [s0] = 128
R13 [t5] = 0
R14 [t6] = 0
R15 [t7] = 0
R16 [s0] = 0
R17 [s1] = 0
R18 [s2] = 0
R19 [s3] = 0
R20 [s4] = 0
R21 [s5] = 0
R22 [s6] = 0
R23 [s7] = 0
R24 [t8] = 0
R25 [t9] = 0
R26 [k0] = 0
R27 [k1] = 0
R28 [gp] = 268468224
R29 [sp] = 2147483000
R30 [sb] = 0
R31 [ra] = 4194328

```

User data segment [10000000]..[10040000]

```

[10000000]..[1000ffff] 00000000
[10010000] 000000005 0000000120 -3 0000000022
[10010010] 0000000017 -1 0000000001 0000000004
[10010020] 0000000032 0000000000 0000000000 0000000000
[10010030]..[1003ffff] 00000000

```

User Stack [7ffffd78]..[80000000]

```

[7ffffd80] 0000000001 2147483078
[7ffffd80] 2147483065 214748336 2147483612 2147483590
[7ffffd90] 2147483552 2147483517 2147483498 2147483483
[7ffffda0] 2147483429 2147483391 2147483327 2147483263
[7ffffdb0] 214748319 2147483183 2147483161 2147483111
[7ffffdc0] 0000000000 0000000000 0000000000 0000000000
[7ffffdd0] 0796227184 0895639884 1835977677 2084037473
[7ffffde0] 1630432377 1490931463 1239213773 19828087378
[7fffff00] 1714385585 1701880175 16472776914 2019705854
[7fffff10] 0795897284 078923080 2002631409 2038051176
[7fffff20] 1785409657 1899311417 202631409 2038051176
[7fffff30] 0059737972 008477561 1852256308 0003101743
[7fffff40] 1313296280 1827951937 1885694308 1476425078
[7fffff50] 1398752088 1238393925 1314866499 1875971937
[7fffff60] 1819380129 1825239497 0778893417 0778923875
[7fffff70] 1970430311 1953574515 1835626611 0892744494
[7fffff80] 0895732084 1969310720 1231577966 1635021678
[7fffff90] 1231381358 1144477052 0226684980 1143816497
[7fffffa0] 07589737241 1111048943 1093676357 200966888
[7fffffb0] 0910443363 0892746111 1392521776 1280066888
[7fffffc0] 11866482381 1973873734 1330118746 079254589
[7fffffd0] 1919251285 1768173427 1970696293 1413566464
[7ffffff0] 1966630152 1647277659 079235856 09880314466
[7ffffff10] 1970430263 1768059695 0892744494 080723599
[7ffffff20] 117987700 1835977677 0000000000 0000000000
[7ffffff30] 1919240742 1836815645 1919877010 1898869621
[7ffffff40] 1232977268 1329791085 1312902477 1338478724
[7ffffff50] 1666949708 08467552110 0003354672 1438472019
[7ffffff60] 1498696018 1397966163 1229868873 0942751044
[7ffffff70] 0003409550 1388275479 1781405757 0007608032

```

User Stack [7ffffd78]..[80000000]

```

[7ffffd80] 0000000001 2147483078
[7ffffd80] 2147483552 2147483357 2147483498 2147483483
[7ffffd90] 2147483349 2147483391 2147483327 2147483263
[7ffffda0] 214748319 2147483183 2147483161 2147483111
[7ffffdb0] 0000000000 0000000000 0000000000 0000000000
[7ffffdc0] 1835237363 1935280623 1781737348 114888923
[7ffffdd0] 1288332617 19862549697 1986622864 1701880175
[7fffffe0] 0795897284 0778923875 1819380129 163479717
[7fffff00] 1751346885 19167644772 202631409 2038051176
[7fffff10] 1735552815 197370606 205451808 1476407354
[7fffff20] 1819380129 1825239497 0778893417 0778923875
[7fffff30] 1970430311 1953574515 1835626611 0892744494
[7fffff40] 0895732084 1969310720 1231577966 1635021678
[7fffff50] 1231381358 1144477052 0226684980 1143816497
[7fffff60] 1313296280 1827951937 1885694308 1476425078
[7fffff70] 1819380129 1825239497 0778893417 0778923875
[7fffff80] 1970430263 1768059695 0892744494 080723599
[7fffff90] 1163447917 1751346805 0694971492 1111638643
[7fffffa0] 1835237363 1935280623 1781737348 114888923
[7fffffb0] 1229213587 0795897284 1819380129 163479717
[7fffffc0] 0893173034 1986622864 1701880175 1835977677
[7fffffd0] 1751346805 197370606 205451808 1476407354
[7ffffff0] 1398752088 1238393925 1314866499 1875971937
[7ffffff10] 1819380129 1825239497 0778893417 0778923875
[7ffffff20] 1970430311 1953574515 1835626611 0892744494
[7ffffff30] 1231381358 1144477052 0226684980 1143816497
[7ffffff40] 0910443363 0892746111 1392521776 1280066888
[7ffffff50] 1498696018 1397966163 1229868873 0942751044
[7ffffff60] 0003409550 1388275479 1781405757 0007608032

```