

The screenshot displays the Immunity Debugger interface. At the top, the menu bar includes File, Edit, View, Project, Flash, Debug, Peripherals, Tools, SVCS, Windows, and Help. Below the menu is a toolbar with various icons for file operations, editing, and debugging. The main window is divided into several panes. On the left, the 'Registers' pane shows a list of registers (R0 through R15) and their current values, all of which are 0. The central pane displays assembly code for a function named 'winnersub'. The code includes instructions for loading values from memory, comparing counters, and branching. The right pane shows the 'Memory' window with a hex dump and its corresponding ASCII representation. The bottom status bar indicates the current instruction address and provides a quick reference for various debugger features like breakpoints, watchpoints, and coverage analysis.

The screenshot displays the Immunity Debugger interface with the following components:

- Top Bar:** Shows the file path `C:\Users\user\Documents\222020\debug\project\review\project\review\Non-Commercialize\license` and standard application menus (File, Edit, View, Project, Flash, Debug, Peripherals, Tools, SVCS, Windows, Help).
- Registers Panel:** Lists registers R0 through R15 and their current values, all pointing to `0:00000007`.
- Disassembly Panel:** Shows assembly instructions for a loop. The instruction at address 101 is highlighted: `bgt eax1 r16 > .branch exit`. Other instructions include `stsb r2, [r1]`, `ldsb r3, [r7]`, `ldsb r4, [r10]`, `cmp r12, r11`, `bgt eax1 r16 > .branch exit`, `b reverse_loop`, `cmp r2, r4`, `bgt homewin /home > away, branch`, `cmp r3, r4`, `blt awaywin /home < away, branch`, `mov r1, eax1`, `stsb r5, [r9]`, `mov r5, r1`, and `stsb r5, [r9]`.
- Command Panel:** Contains the command `!include "C:\Users\user\Documents\222020\debug\review\Non-Commercialize\license"` and the output `PC = 0x00000000 //this is the program counter starts`.
- Memory Panel:** Shows a memory dump starting at address `0x00000000`. The first row of data is `0x00000000: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00`.
- Bottom Bar:** Includes a status bar with `ASION BreakDisable BreakEnable BreakKill BreakList BreakAddr BreakAddress COVERAGE COVOTITLE DEFINE DIR Display Enter EYALUATE` and a `Call Stack - Local` button.

C:\Users\yz01\Desktop\elec2220\big_review\project final\big_review - [Non-Commercial Use License]

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

Registers

Register	Value
R0	0x00000002
R1	0x20000002
R2	0x00000008
R3	0x00000002
R4	0x00000008
R5	0x000000FF
R6	0x20000071
R7	0x2000004F
R8	0x20000020
...	...

big_review_1 debug.ini

```
91 ldrb r2,[r10,r0]; ldrb rx=score = representation
92 stxb r2,[r0];representation to scoreboard
93
94 ldrb r3,[r7]; upload Unpacked_away value to
95 ldrb r4,[r10,r3]; ldrb rx=score = representation
96 stxb r4,[r0];representation to scoreboard second byte
97 stxb r4,[r6];store r4 to bonus_list
98 stxb r5,[r8];store r5 to bonus_list second byte
99 bl winner_sub
100 cmp r12,r11;compare counter and length of unpacked_home
101 bgt exit;if >,branch exit
102 b sevenseq_loop
103
104 winner_sub
105 cmp r2,r4
106 bgt homewin;home > away, branch
107 cmp r2,r4
108 blt awaywin;home < away,branch
109 mov r5,#0xFF; -1
110 stxb r5,[r9]; store to winning
111 b back
112 homewin
113 mov r5,#0; 0
114 stxb r5,[r9]
115 b back
116 awaywin
117 mov r5,#1; 1
118 stxb r5,[r9]
119
```

Command

```
Include "C:\Users\yz01\Desktop\elec2220\big_review\...\debug.ini"
MPSR = 0x01000000 //this is thumb mode
PC = 0x08000000 //this is the program counter starts
//X CHSR 0x02000000 = 1, 10, 2, -4 JE CHSR is gdb
E INT 0x02000000 = 0x12345678, 0x7123123, 0
E INT 0x02000010 = 0x02102736, 0x03491238, 0
E char 0x02000020 = 0x3F,0x3,0x5B,0x4F,0x64,0x6D,0x7D,0x07,0x7F,0x7F
IF = 0x020001000
```

Memory 1

Address	0x20000020
0x20000020: 18 18 FF 3F 03 5B 4F 64 6D 7D 07 7F 7F 01 02 03 04 05 06 07 08 07 07 01 02 03 04 05 06 00 00 00 00	
0x20000045: 00 00 00 00 00 00 00 00 09 02 01 00 02 07 03 06 08 03 04 09 01 02 03 08 00 00 00 00 00 00 00 00 00	
0x2000004A: 00 00 00 77 03 18 18 00	
0x2000008F: 00	
0x20000084: 00	
0x20000029: 00	
0x200000FE: 00	
0x20000123: 00	
0x20000141: 00	
0x2000012B: 00	

Assign BreakDisable BreakEnable BreakFill BreakList BreakLet BreakLet BreakAddress COVTOFILE DEFINE DIR Display Enter Evaluate

Simulation 11:0.00005525 sec L100-C1 GAP NEUM SCRL OVR RW

main,

bl home_score,

bl away_score,

home_score, away_score
push {lr} // ~~has~~ same operating logic as home_score

ldr homescores,

ldr r, unpacked_home

mov r, 0,

mov r, 28,

home loop first value

upload homescores to register

right shift 7 byte, through tcr 28

28-7, next loop shift 6 byte

only save tcr; clear other bits except lsb

store lsb into unpacked_home array.

index score

count +1

~~compare~~ compare count length = 8, reset r, if not, loop again.

check next byte of homescores, if 0 exit home loop, if not 0, loop again

exit

pop {lr}

bx lr

homescores dcd 0, 0, 0, 0

awayscores dcd 0, 0, 0, 0,

scoreboard dcb 0, 0,

winning dcb 0,

scoresequat dcb 0x3F, 0x03, 0x5B, 0x4F, ...

unpacked_home space 32,

unpacked_away space 32.

sub sevensesq

loop ~~ltx~~ ~~counter~~ | counter+1, push {lr}

r1 = score.

ldr rx = sevensesqual

ldr b, rx + score \Rightarrow representation.

bl winnersub

cmp ~~bx~~ counter, length of unpacked - home.

If counter > unpacked, branch exit.

b loop

winnersub

If home > away, ^{branch} home homewin

If home < away, branch awaywin

~~more~~ \Rightarrow

mov rx, -1

~~back~~ bx lr

homewin

mov rx, 0,

bx lr

awaywin

mov rx, 1

bx lr

exit

pop {lr}

bx lr

bonus

~~set~~ ~~set~~ ~~put~~ ~~put~~

r_x = representation of homescores.

r_y = ~~anayscores~~ representation of anayscores

r_z = bonus_list

~~put~~ ~~set~~ strb r_y , [r_z]#1

strb r_x , [r_z]#1

~~put~~ ~~set~~

ldr r_0 , [r_z , index]

~~add~~ r index + 2

lsl r_0 , #16

lsr r_0 , #16.

str r_0 [bonus, Index]

Index + 4

cmp r_0 , #0

~~b loop~~, beq end

b loop.

end*

bx lr

