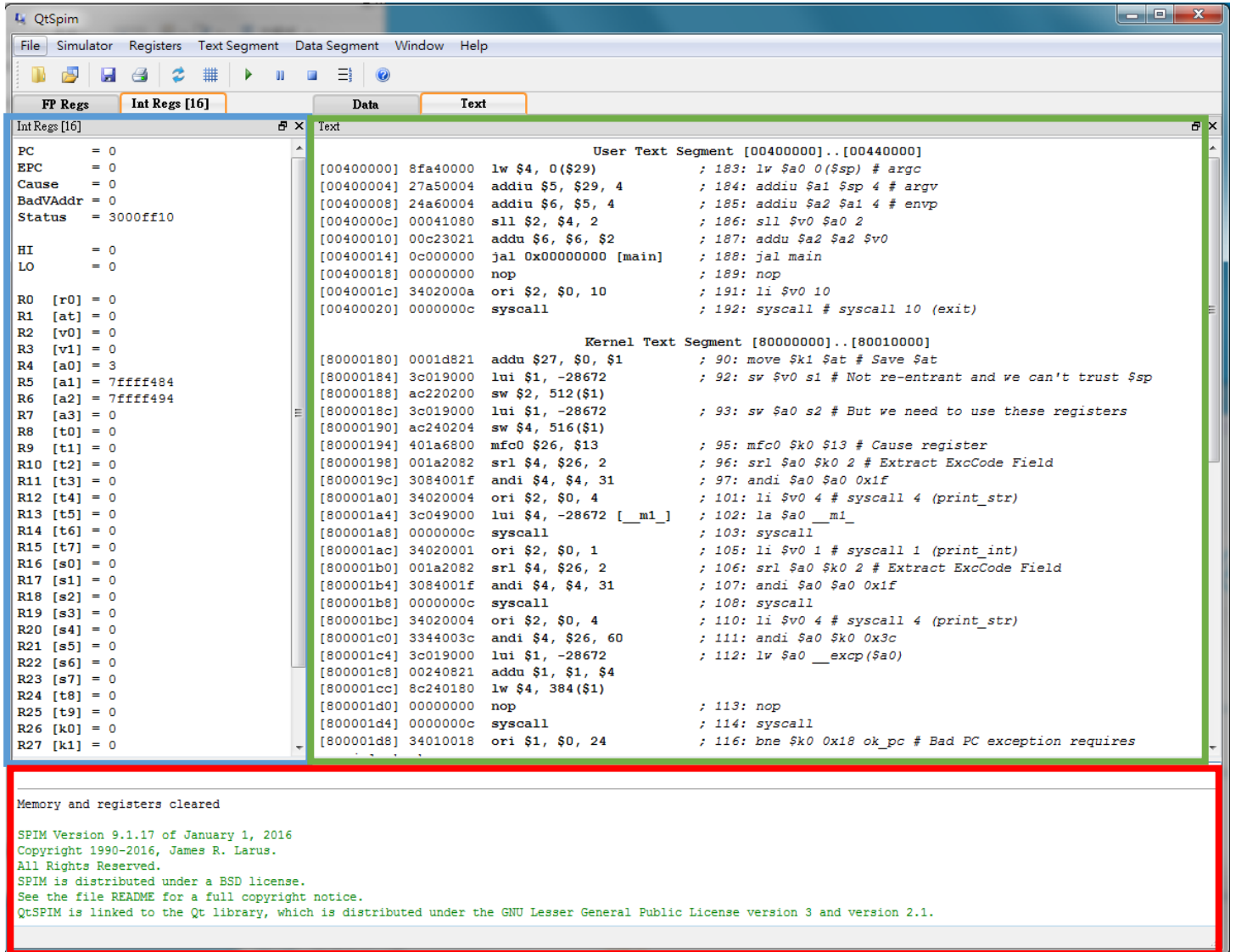


QtSPIM MIPS Simulator 使用說明

1. 先去以下載點下載並安裝

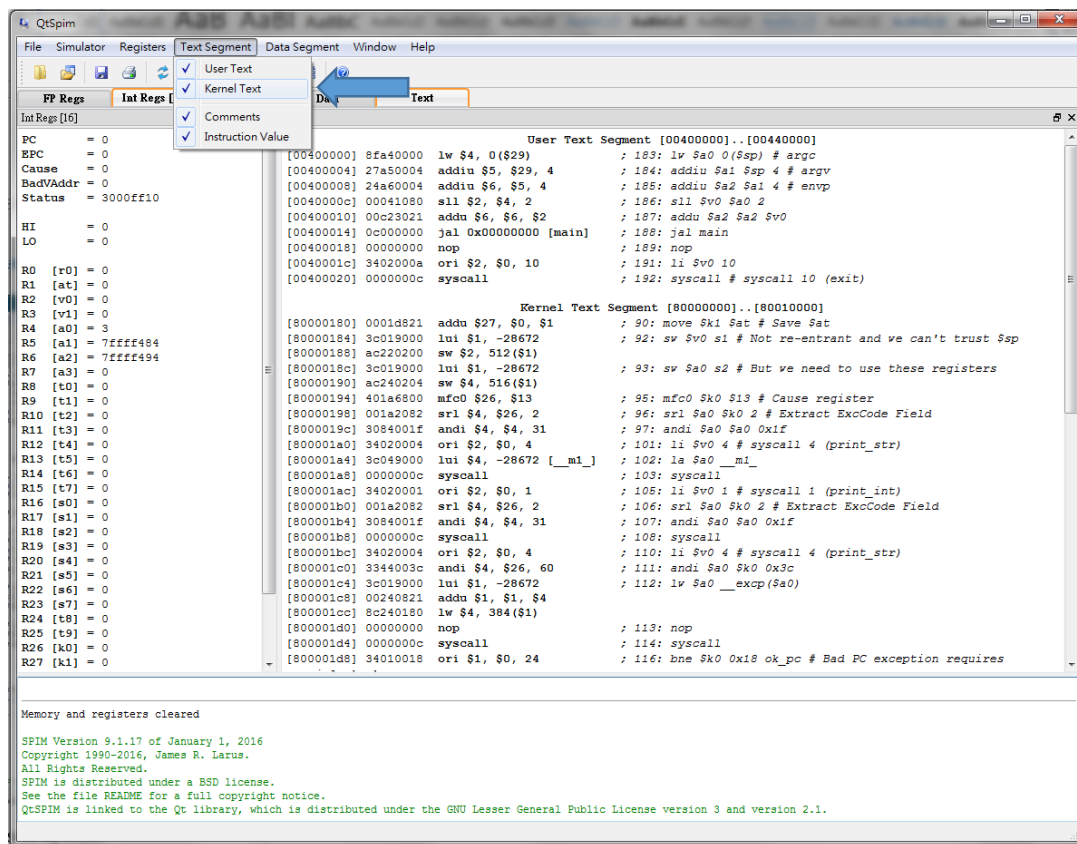
<https://sourceforge.net/projects/spimsimulator/?source=navbar>


2. 開啟 QtSPIM

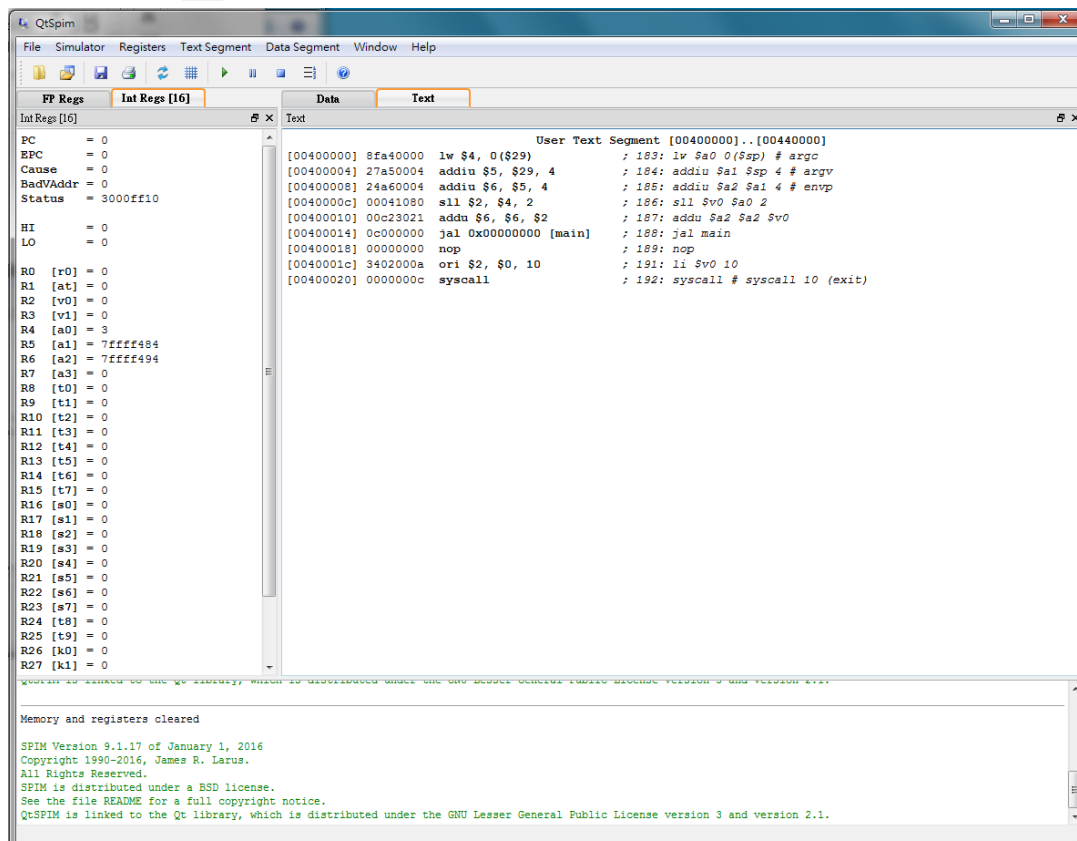


左邊為 Register Part，右邊為 Text & Data segment，下方為 SPIM log output


3. 環境 setup: 點選上方 Text Segment > 取消勾選 Kernel Text
(我們不需要看 kernel 的部分)

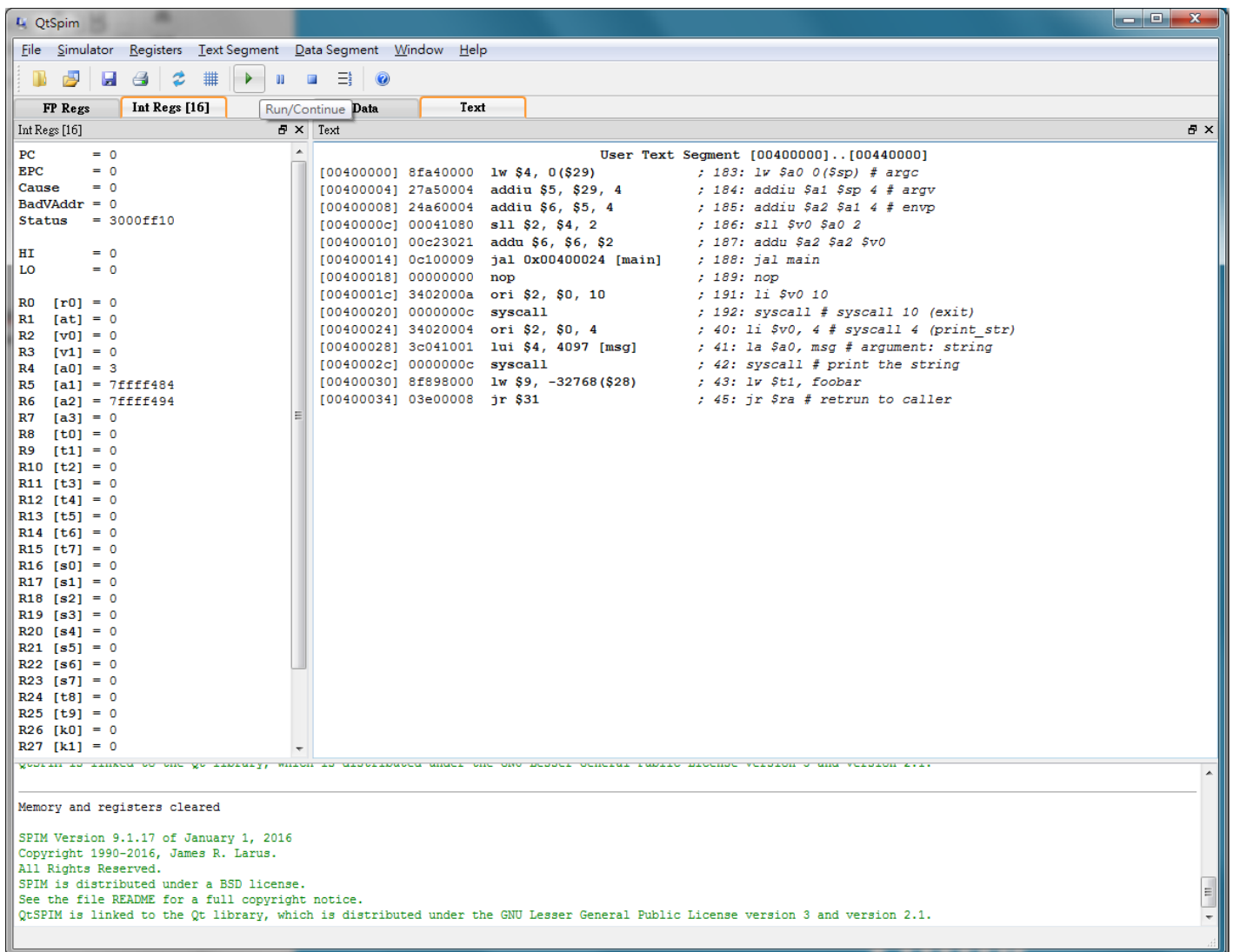


— 接著按  Reinitialize Simulator 這個東東，就會剩下 User Text

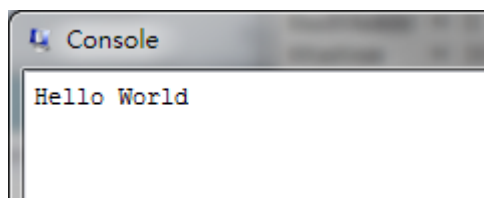


4. 開始 load assembly code: (以 helloworld.s 為例)

點選  Reinitialize and load file，選擇 C:\Program Files (x86)\QtSpim(安裝路徑)內的 helloworld.s，Simulator 就會將 MIPS code 轉換成 machine code



接著點選 Run/Continue，Console 就會出現結果



比照此方法 Load 自己的 assembly code 即可

※編輯 code 需用其他 editor，如 notepad++

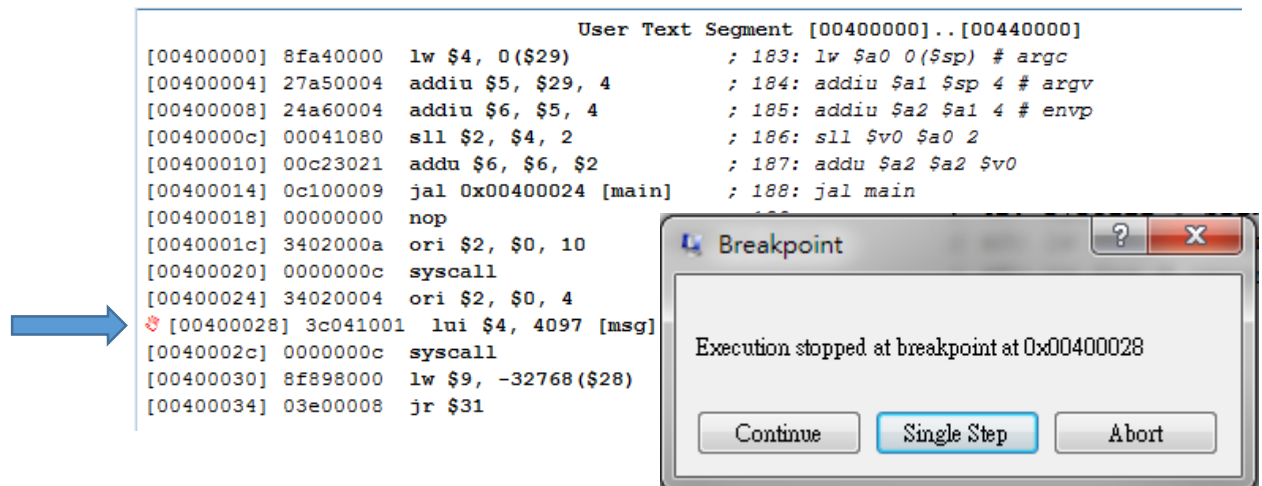
5. Debug 方法

I. Single Step:

Load assembly code 後，可以利用  Single Step，一步一步執行各個 instruction，配合觀看左方的 Register 的值，以及 Data segment 內各個記憶體位址的值，來驗證 code 正確性

II. Set Breakpoint

對想要 Break 的 instruction 右鍵>選擇 Set Breakpoint，然後點下 Run/continue，程式就會執行到 Breakpoint 並噴出訊息：



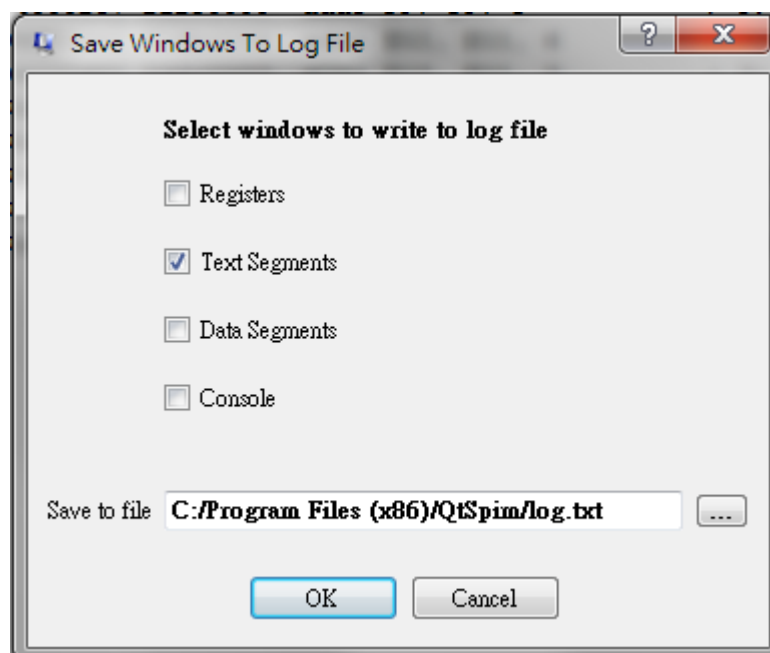
The screenshot shows the QtSpim assembly viewer with the 'User Text Segment' selected. The assembly code is displayed in two columns. A blue arrow points to the instruction at address 00400028: `lui $4, 4097 [msg]`. A 'Breakpoint' dialog box is open, displaying the message 'Execution stopped at breakpoint at 0x00400028'. The dialog has three buttons: 'Continue', 'Single Step' (which is highlighted), and 'Abort'.

```

User Text Segment [00400000]..[00440000]
[00400000] 8fa40000 lw $4, 0($29)           ; 183: lw $a0 0($sp) # argc
[00400004] 27a50004 addiu $5, $29, 4         ; 184: addiu $a1 $sp 4 # argv
[00400008] 24a60004 addiu $6, $5, 4         ; 185: addiu $a2 $a1 4 # envp
[0040000c] 00041080 sll $2, $4, 2           ; 186: sll $v0 $a0 2
[00400010] 00c23021 addu $6, $6, $2         ; 187: addu $a2 $a2 $v0
[00400014] 0c100009 jal 0x00400024 [main]   ; 188: jal main
[00400018] 00000000 nop
[0040001c] 3402000a ori $2, $0, 10
[00400020] 0000000c syscall
[00400024] 34020004 ori $2, $0, 4
[00400028] 3c041001 lui $4, 4097 [msg]
[0040002c] 0000000c syscall
[00400030] 8f898000 lw $9, -32768($28)
[00400034] 03e00008 jr $31
```

6. Output machine code

- 點選  勾選 Text Segment
- 選擇要存的 folder 並命名 xxx.txt



User Text :

PC	Machine code(hex)	assembly	comment
[00400000]	8fa40000	lw \$4, 0(\$29)	; 183: lw \$a0 0(\$sp) # argc
[00400004]	27a50004	addiu \$5, \$29, 4	; 184: addiu \$a1 \$sp 4 # argv
[00400008]	24a60004	addiu \$6, \$5, 4	; 185: addiu \$a2 \$a1 4 # envp
[0040000c]	00041080	sll \$2, \$4, 2	; 186: sll \$v0 \$a0 2
[00400010]	00c23021	addu \$6, \$6, \$2	; 187: addu \$a2 \$a2 \$v0
[00400014]	0c100009	jal 0x00400024 [main]	; 188: jal main
[00400018]	00000000	nop	; 189: nop
[0040001c]	3402000a	ori \$2, \$0, 10	; 191: li \$v0 10
[00400020]	0000000c	syscall	; 192: syscall # syscall 10 (exit)
[00400024]	2108000c	addi \$8, \$8, 12	; 4: addi \$t0, \$t0, 12
[00400028]	21290005	addi \$9, \$9, 5	; 5: addi \$t1, \$t1, 5
[0040002c]	216b0004	addi \$11, \$11, 4	; 7: addi \$t3, \$t3, 4
[00400030]	296c0003	slti \$12, \$11, 3	; 9: slti \$t4, \$t3, 3 #if(\$t3
[00400034]	11800003	beq \$12, \$0, 12 [LABEL-0x00400034]	
[00400038]	01095020	add \$10, \$8, \$9	; 12: add \$t2, \$t0, \$t1
[0040003c]	08100011	j 0x00400044 [QUIT]	; 13: j QUIT
[00400040]	01095022	sub \$10, \$8, \$9	; 15: sub \$t2, \$t0, \$t1
[00400044]	03e00008	jr \$31	; 18: jr \$ra # retrun to caller

7. [Optional settings] Enable Delayed Branches/Enable Delayed Loads

由於 MIPS pipeline CPU 的 Branch 以及 LW 指令會 delay 一個 cylce

若需要實際將 Branch 翻譯成機械碼為正確的 address 值(PC=PC+4+imm)，就將 **Enable Delayed Branches** 勾起，否則 address 值會是 PC=PC+imm 而已

*此動作不影響撰寫 assembly code

