Homework4 不对齐的LDR与STR设计

Problem

为了节省内存，避免内存中出现空洞，常采用不对齐的方式分配内存。然而，不对齐的数据，无法用指令“LDR”和“STR”正确从内存装载和写入内存。

（1）请编写两条宏定义指令“LDRUA”和“STRUA”，能够将32位数据正确从内存装载和写入内容，而不论数据是否对齐。

（2）请用编写程序，用实际数据验证两条自定义指令。

Answer

本次作业在Keil环境下测试，针对输入的32位数据进行不对齐地址测试，通过宏定义“LDRUA”和“STRUA”，结合LDRB和STRB单字节指令构造

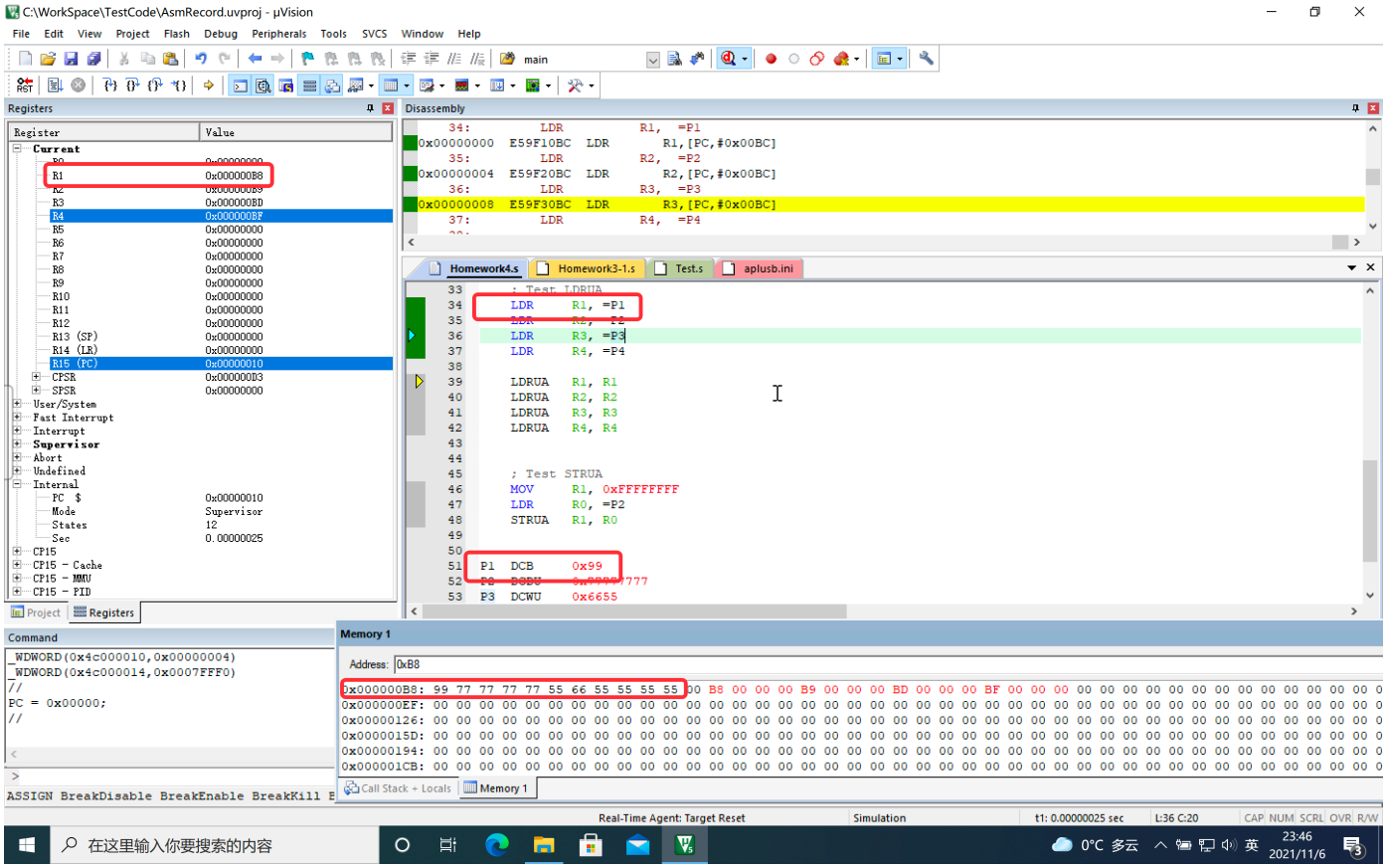
代码

```
; implement LDRUA
MACRO
LDRUA $RD,$RS
LDRB  v2, [$RS]
LDRB  v3, [$RS, #1]
LDRB  v4, [$RS, #2]
LDRB  v5, [$RS, #3]

ADD   v6, v2, v3, LSL #8
ADD   v6, v6, v4, LSL #16
ADD   v6, v6, v5, LSL #24

MOV   $RD,v6
MEND

; implement STRUA
MACRO
STRUA $RD,$RS
MOV   v2, $RD
STRB  v2, [$RS]
LSR   v2, #8
STRB  v2, [$RS,#1]
LSR   v2, #8
STRB  v2, [$RS,#2]
LSR   v2, #8
STRB  v2, [$RS,#3]
MEND

AREA     ARMex, CODE, READONLY
```

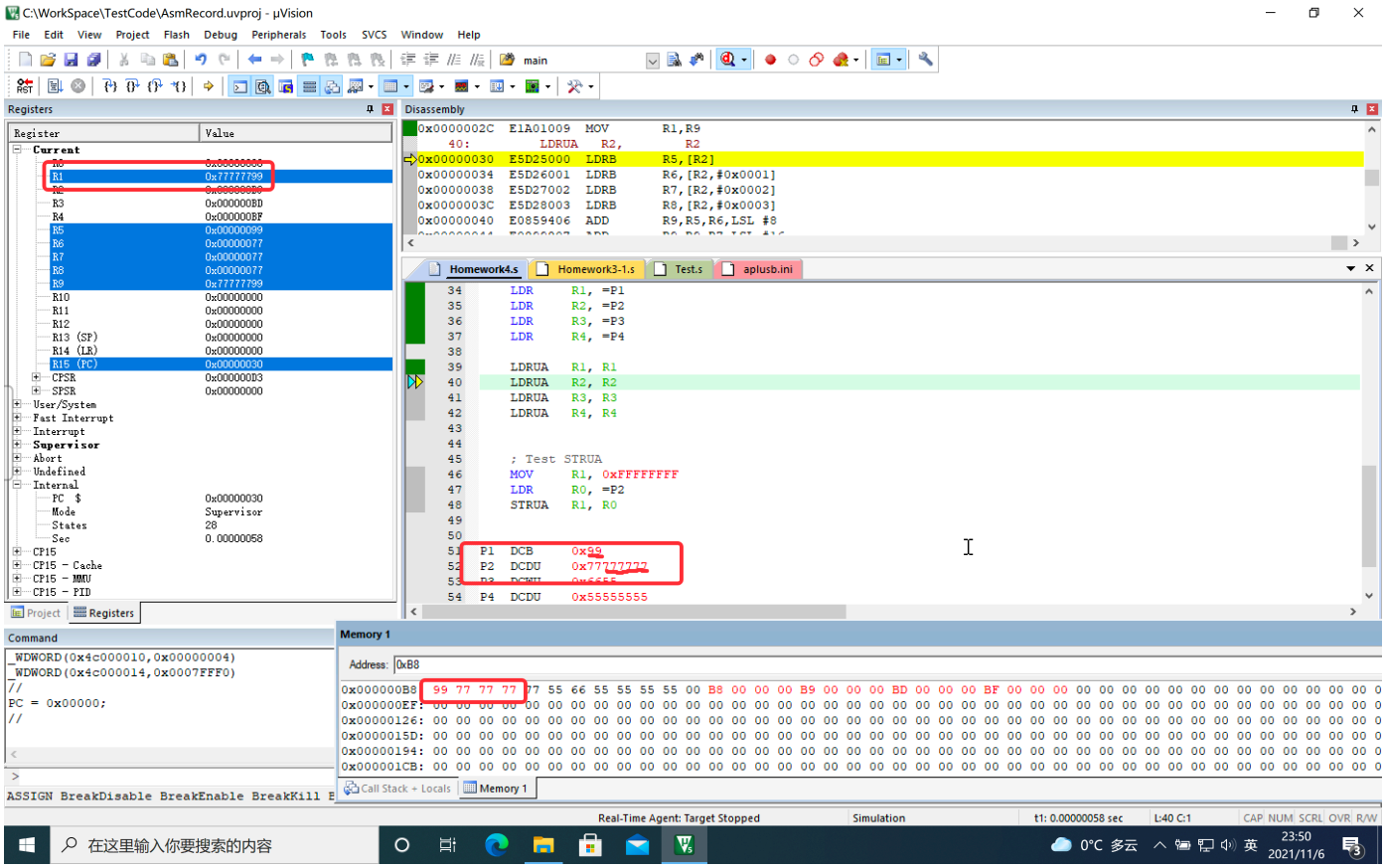
```
    ENTRY
  CODE32

  ; Test LDRUA
  LDR   R1, =P1
  LDR   R2, =P2
  LDR   R3, =P3
  LDR   R4, =P4

  LDRUA R1, R1
  LDRUA R2, R2
  LDRUA R3, R3
  LDRUA R4, R4


  ; Test STRUA
  MOV   R1, 0xFFFFFFFF
  LDR   R0, =P2
  STRUA R1, R0


P1  DCB   0x99
P2  DCDU  0x77777777
P3  DCWU  0x6655
P4  DCDU  0x55555555

  END
```

调试

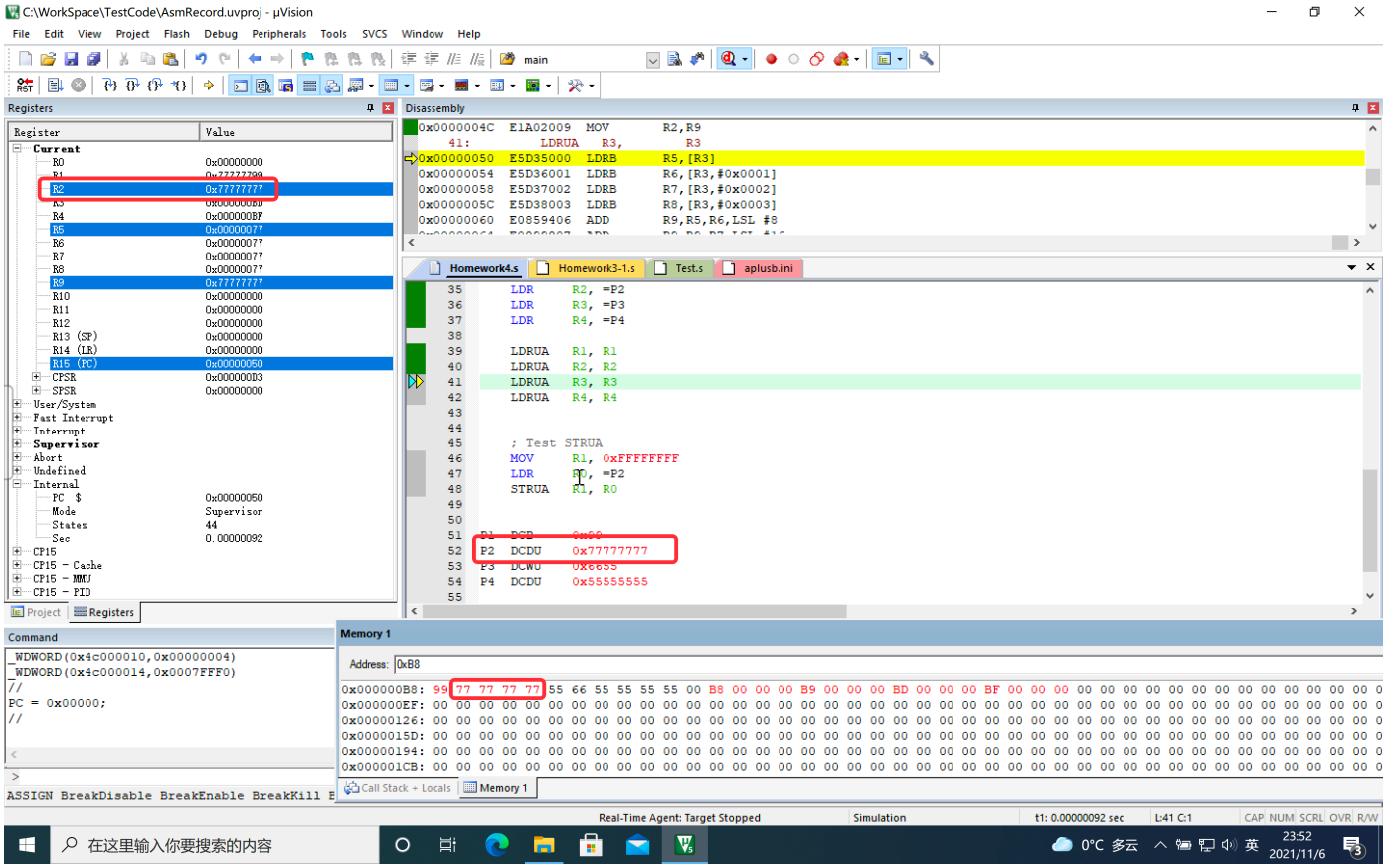
- 首先初始化参数，读入mem中地址设置的数值，从调试结果来看，初始读入数据的地址是0xB8，由于紧接着的内存分别按照1个字节，4个字节，2个字节，4个字节分配，因此R2，R3，R4存储的地址为0xB9，0xBD，0xBF



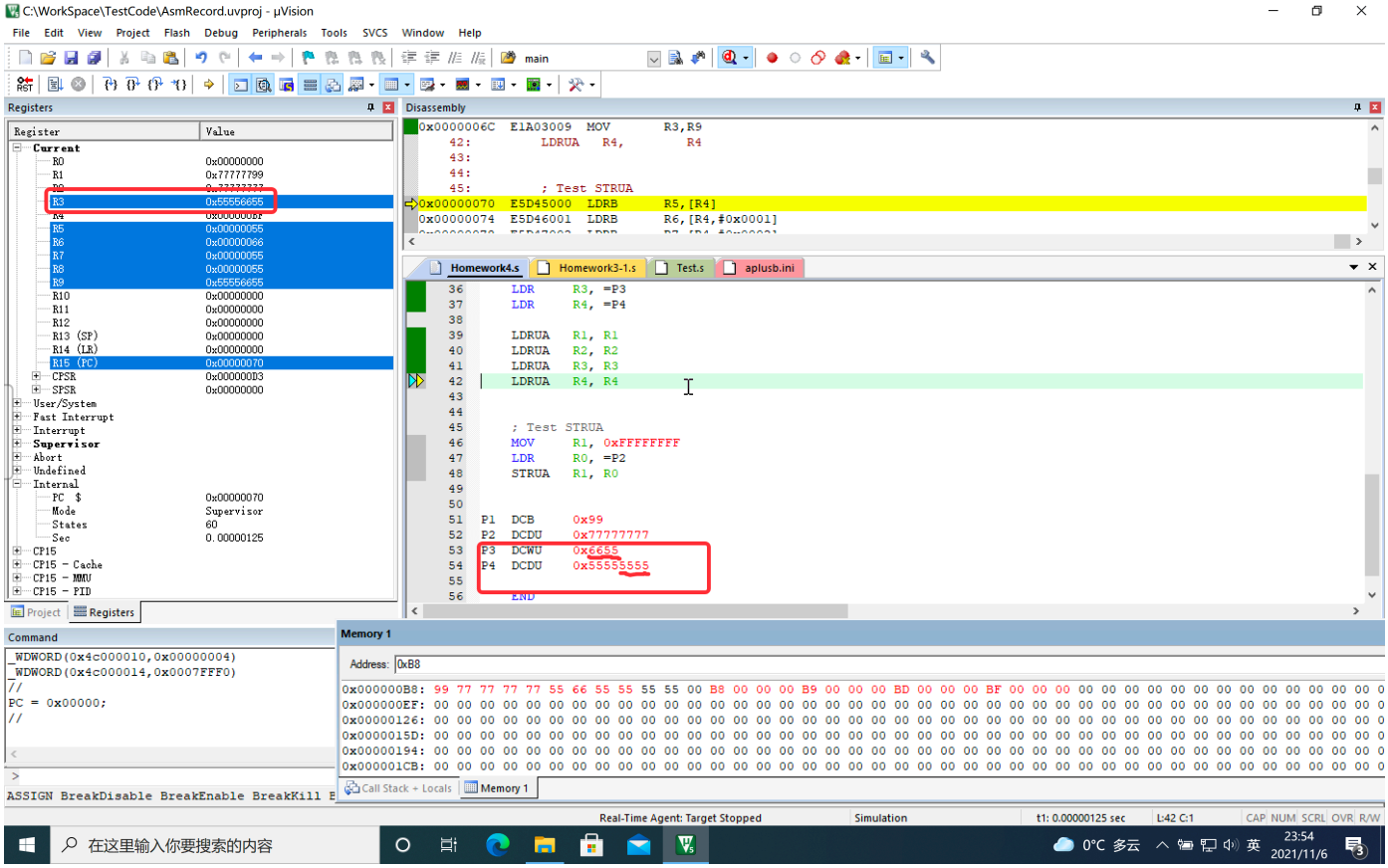
- 随后读入P1标签的数据到寄存器R1



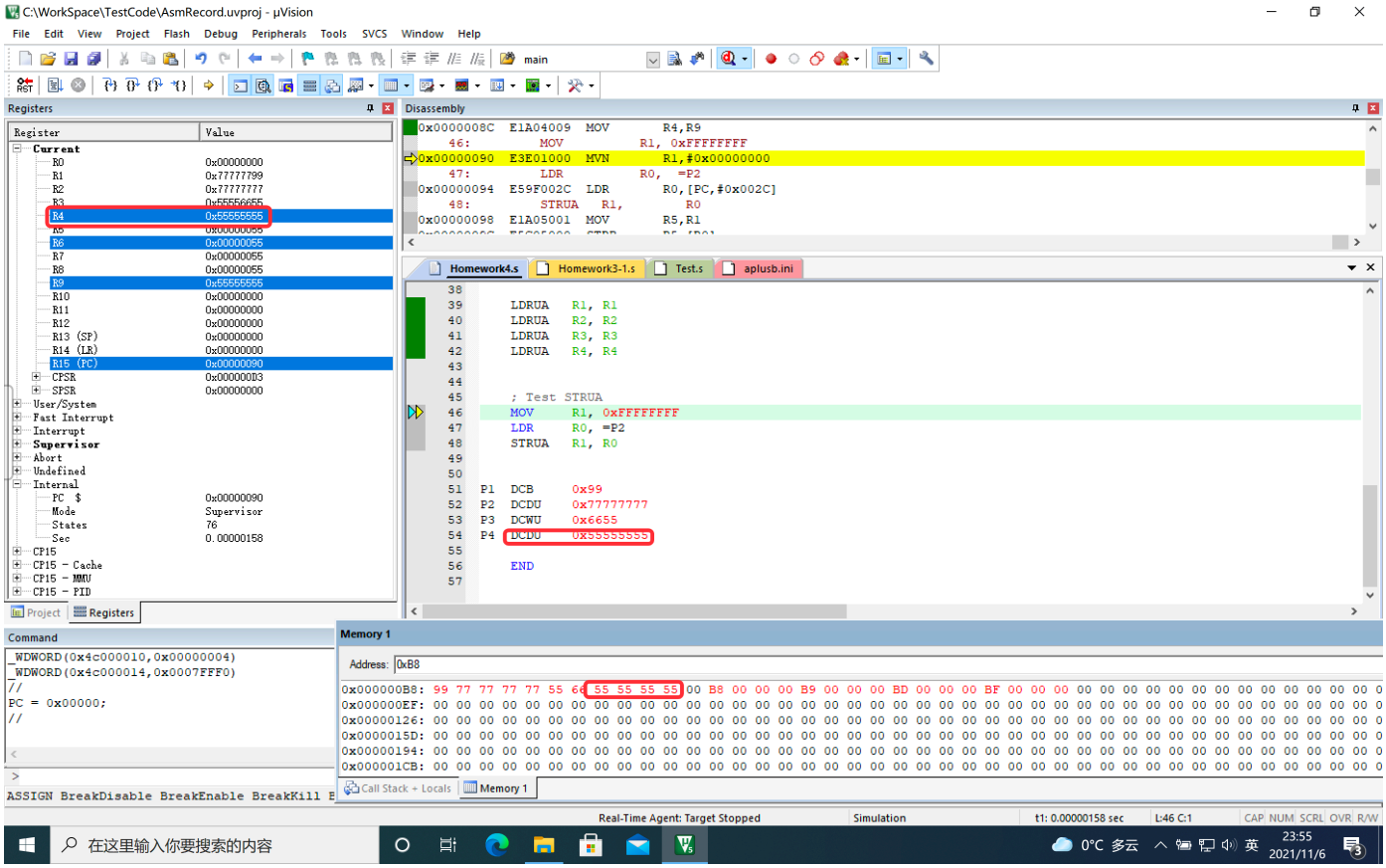
- 读入P2标签的数据到寄存器R2，此时读入的地址是0xB9，不是4的倍数，但是依旧读入到寄存器R2，说明LDRUA的逻辑没有问题



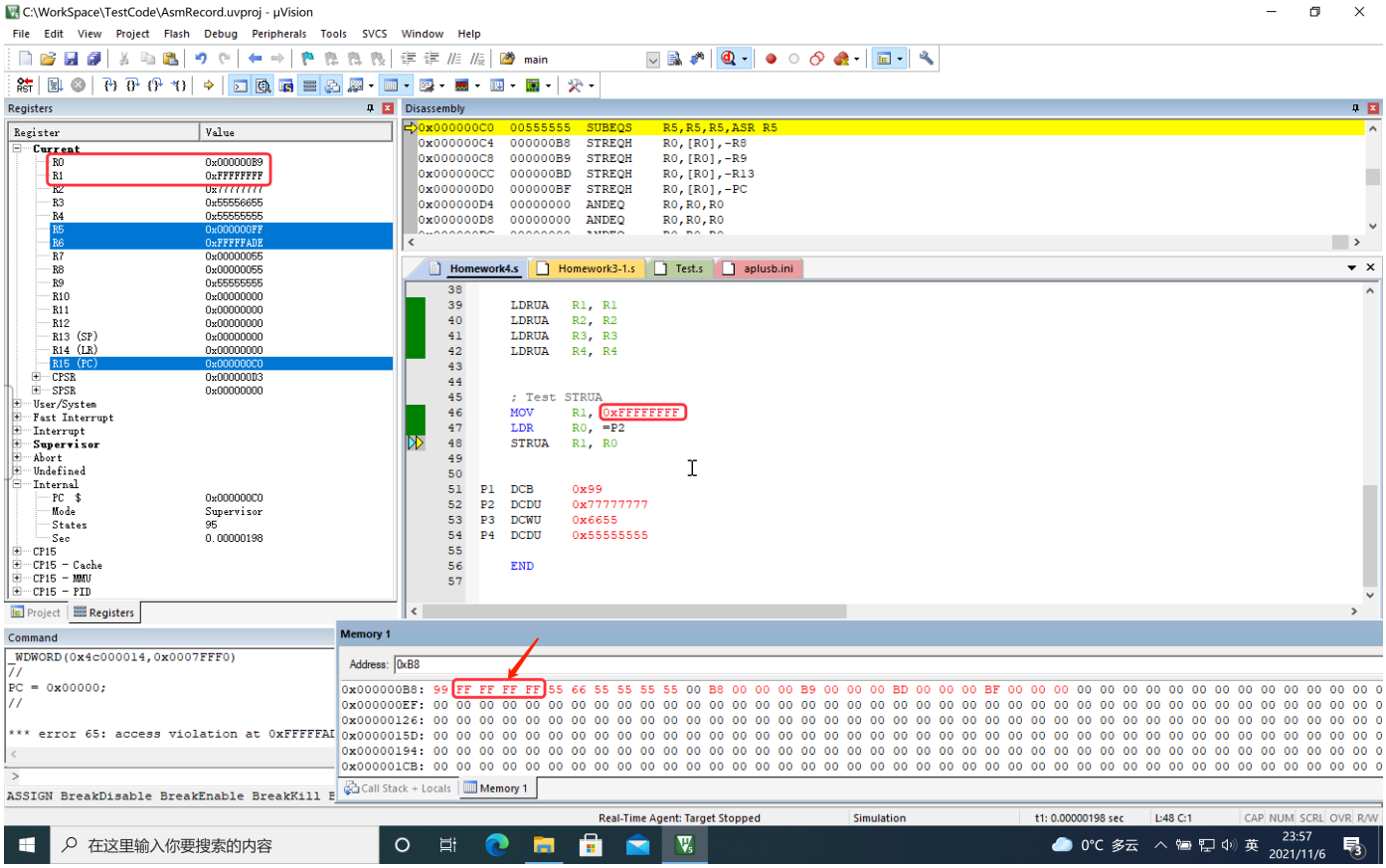
- 读入P3标签的数据到寄存器R3



- 读入P4标签的数据到寄存器R4



- 下面，测试存储到内存中，我们选取读到P2标签的位置，该地址是0xB9，不是4的倍数，没有对齐，成功写入到内存，说明STRUA指令逻辑没有问题



注意事项

- 对于**写入**操作得在Option->Debug->Init中添加初始化配置文件，配置文件内容：

```
MAP 0x000000, 0x0ffff  exec write read
MAP 0x40000000, 0x40010000 write read
MAP 0x53000000, 0x53001000 write read
MAP 0x48000000, 0x48001000 write read
MAP 0x4C000000, 0x4C001000 write read
MAP 0x50000000, 0x50001000 write read
//
_WDWORD(0x4c000000,0x00ffffff)
_WDWORD(0x4c000004,0x00000000)
_WDWORD(0x4c000008,0x0005C080)
_WDWORD(0x4c00000C,0x00028080)
_WDWORD(0x4c000010,0x00000004)
_WDWORD(0x4c000014,0x0007FFF0)
//
PC = 0x00000;
```

- LDR系列第二个参数需要加[]，否则会报错：

```
error: A1114E: Expected register relative expression
```

- 对于需要穿寄存器的时候不要弄混淆传递了寄存器中的值，否则会报错：

```
error: A1647E: Bad register name symbol, expected Integer register
```