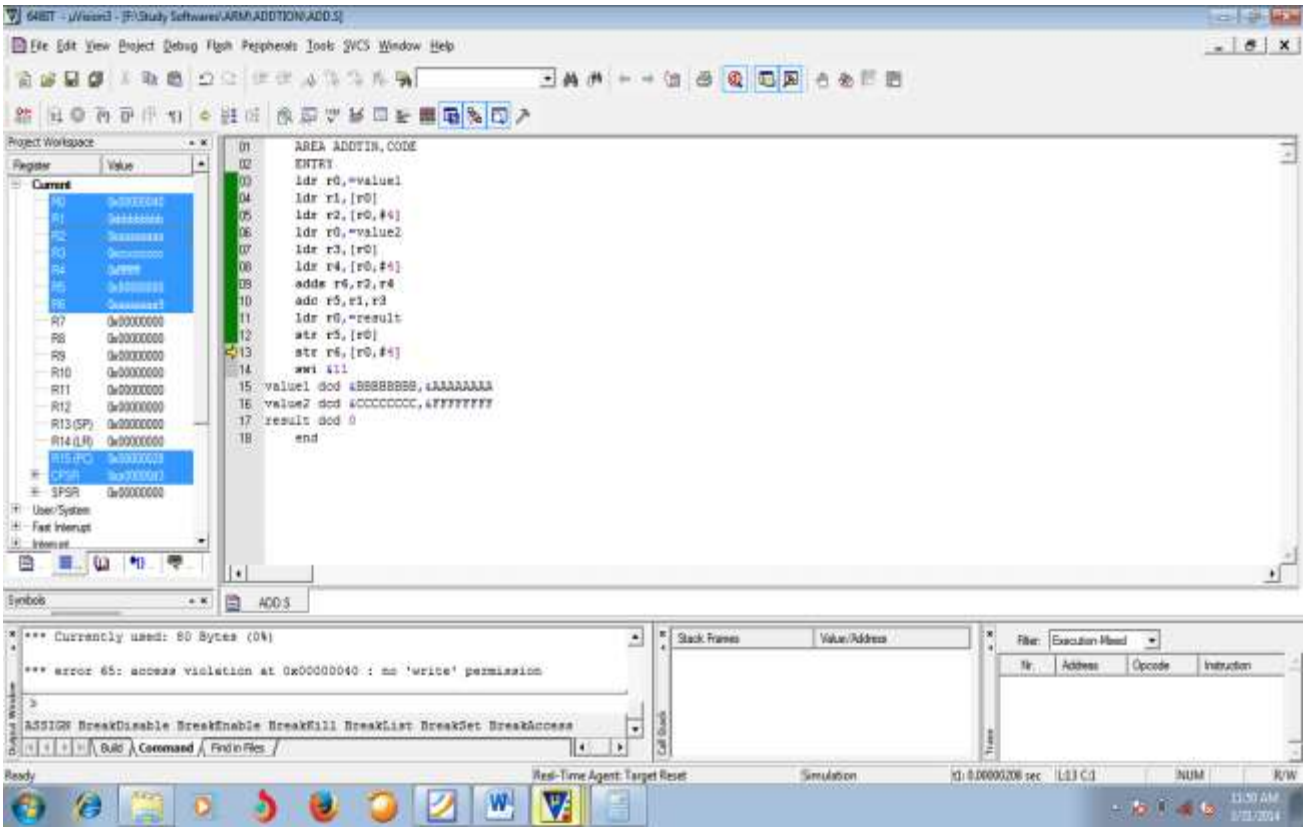
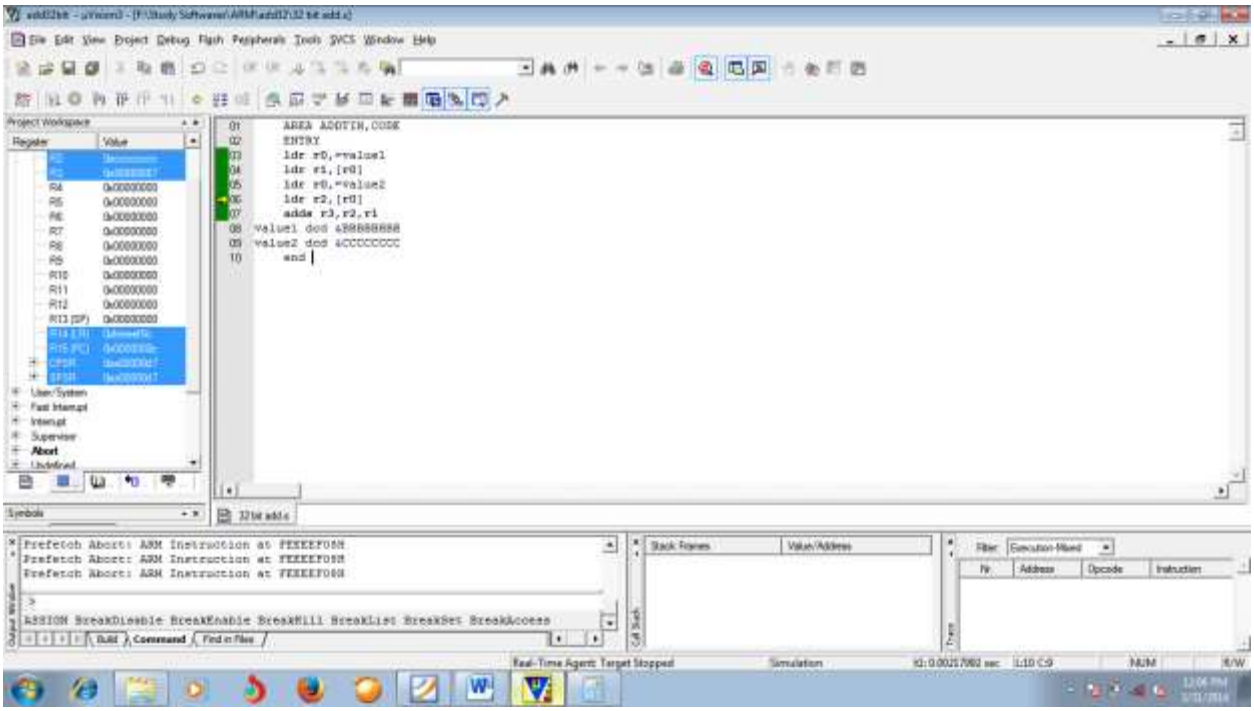
1 write an ALP for addition two 64 bit numbers .

```
        AREA ADDTIN,CODE
        ENTRY
        ldr r0,=value1
        ldr r1,[r0]
        ldr r2,[r0,#4]
        ldr r0,=value2
        ldr r3,[r0]
        ldr r4,[r0,#4]
        adds r6,r2,r4
        adc r5,r1,r3
        ldr r0,=result
        str r5,[r0]
        str r6,[r0,#4]
        swi &11
value1 dcd &BBBBBBBB,&AAAAAAAA
value2 dcd &CCCCCCCC,&FFFFFFFF
result dcd &0
```

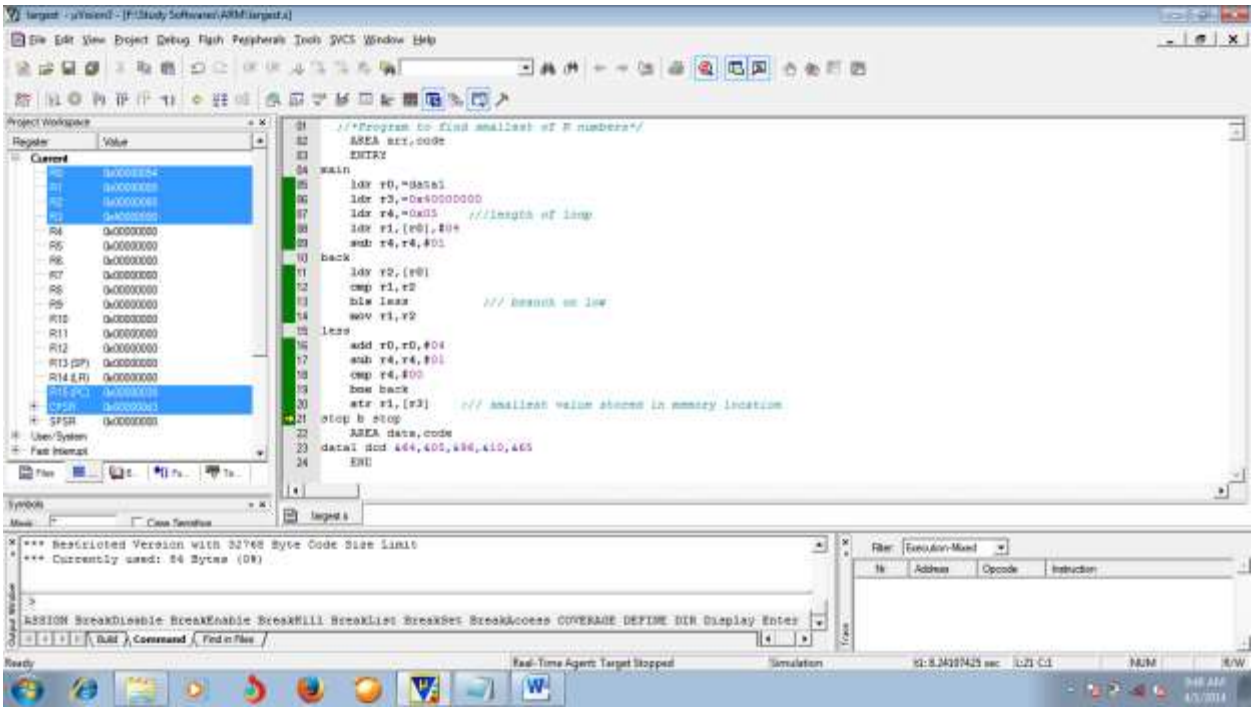


2 write an ALP for addition two 32 bits numbers .

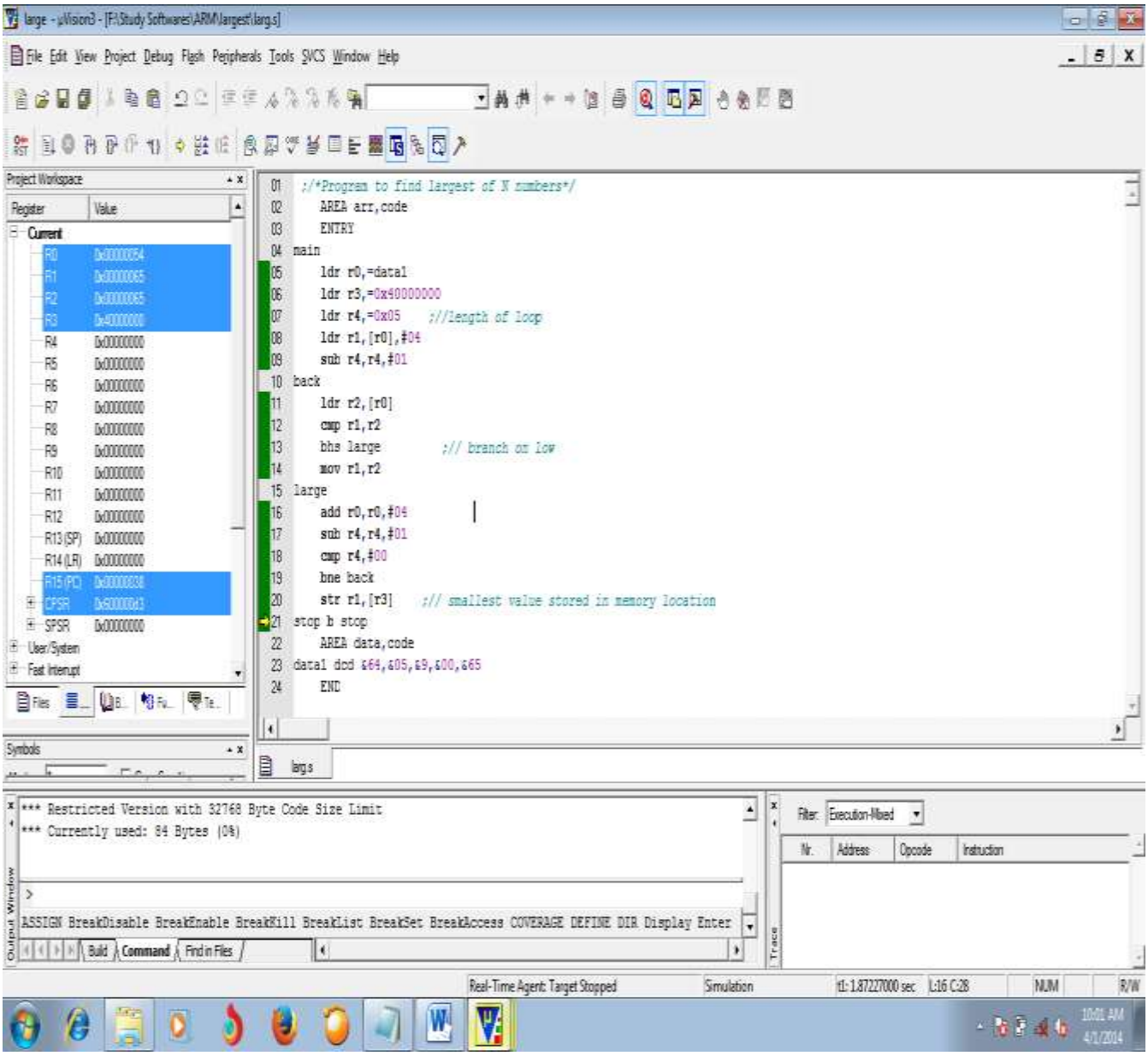
```
        AREA ADDTIN,CODE
        ENTRY
        ldr r0,=value1

        ldr r1,[r0]
        ldr r0,=value2
        ldr r2,[r0]
        adds r3,r2,r1
value1 dcd &BBBBBBBB
value2 dcd &CCCCCCCC
        end
```


File Edit View Project Debug Flash Peripherals Tools SVCS Window Help
Project Workspace
Register
Value
01 AREA ADDTIN,CODE
02 ENTRY
03 ldr r0,=value1
04 ldr r1,[r0]
05 ldr r0,=value2
06 ldr r2,[r0]
07 adds r3,r2,r1
08 value1 dcd &88888888
09 value2 dcd &CCCCCCCC
10 end
Prefetch Abort: ARM Instruction at FEEEEF08H
Prefetch Abort: ARM Instruction at FEEEEF08H
Prefetch Abort: ARM Instruction at FEEEEF08H
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess
Build Command Find in Files
Stack Frames
Value/Address
Filter:
Nr
Address
Opcode
Instruction
Real-Time Agent: Target Stopped
Simulation
L:10 C:9
NUM
R/W

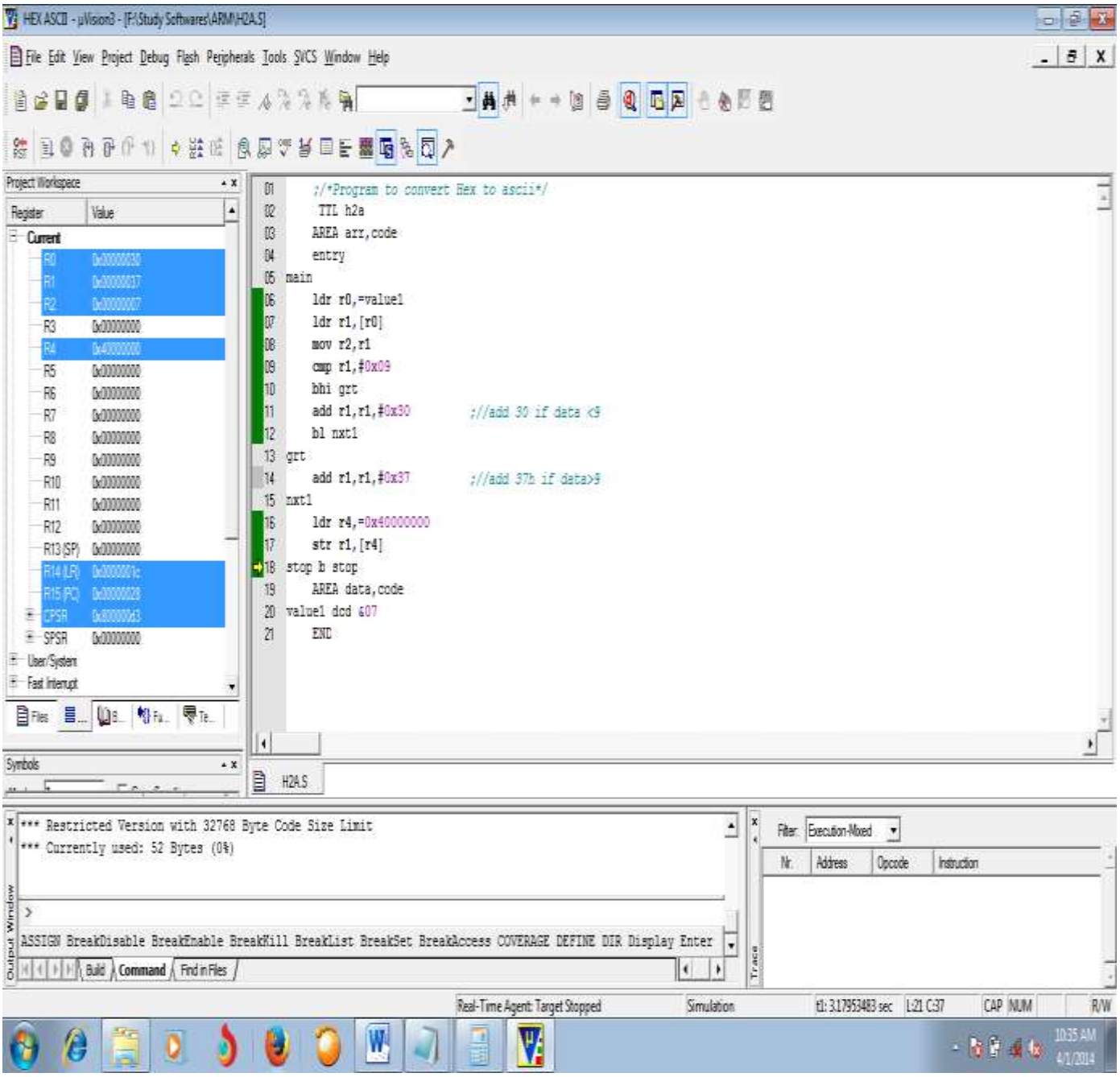
```
;/*Program to find smallest of N numbers*/
        AREA arr,code
        ENTRY
main
        ldr r0,=data1
        ldr r3,=0x40000000
        ldr r4,=0x05    ;//length of loop
        ldr r1,[r0],#04
        sub r4,r4,#01
back
        ldr r2,[r0]
        cmp r1,r2
        bls less            ;// branch on low
        mov r1,r2
less
        add r0,r0,#04
        sub r4,r4,#01
        cmp r4,#00
        bne back
        str r1,[r3]          ;// smallest value stored in memory location
stop b stop
        AREA data,code
data1 dcd &64,&05,&96,&10,&65
        END
```


//*Program to find smallest of N numbers*//
AREA arr,code
ENTRY
main
ldr r0,=data1
ldr r3,=0x40000000
ldr r4,=0x05 ///length of loop
ldr r1,[r0],#04
sub r4,r4,#01
back
ldr r2,[r0]
cmp r1,r2
blo less /// branch on low
mov r1,r2
less
add r0,r0,#04
sub r4,r4,#01
cmp r4,#00
bne back
str r1,[r3] /// smallest value stored in memory location
stop b stop
AREA data,code
data1 dcd 444,405,498,410,465
END
*** Restricted Version with 32768 Byte Code Size Limit
*** Currently used: 84 Bytes (0%)
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE DEFINE DIR Display Enter

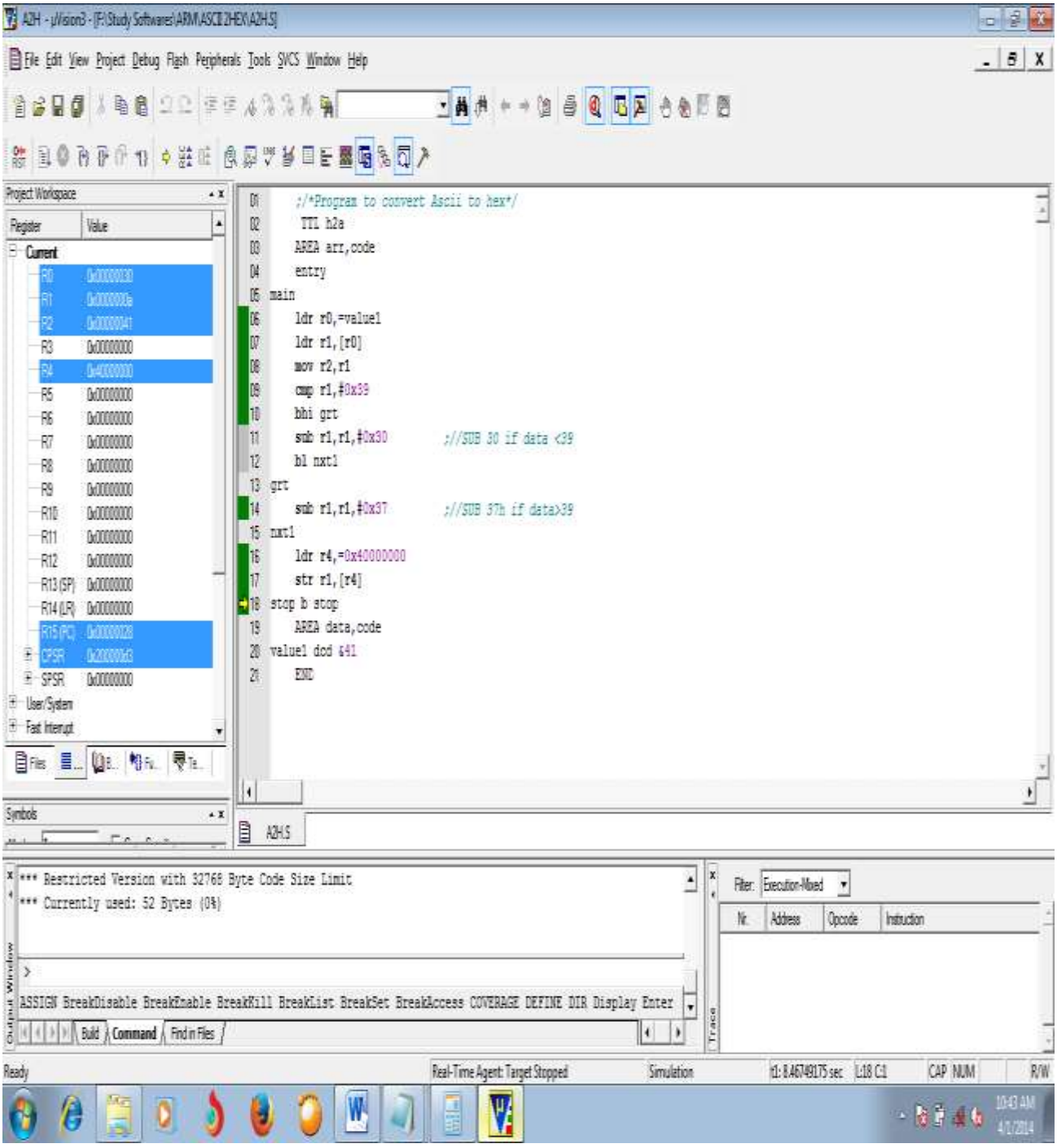
```
;/*Program to find largest of N numbers*/
        AREA arr,code
        ENTRY
main
        ldr r0,=data1
        ldr r3,=0x40000000
        ldr r4,=0x05    ;//length of loop
        ldr r1,[r0],#04
        sub r4,r4,#01
back
        ldr r2,[r0]
        cmp r1,r2
        bhs large                   ;// branch on low
        mov r1,r2
large
        add r0,r0,#04
        sub r4,r4,#01
        cmp r4,#00
        bne back
        str r1,[r3]          ;// smallest value stored in memory location
stop b stop
        AREA data,code
data1 dcd &64,&05,&9,&00,&65
        END
```


large - µVision3 - [F:\Study Softwares\ARM\largest\larg.s]
File Edit View Project Debug Flash Peripherals Tools SVCS Window Help
Project Workspace
Register
Value
Current
R0 0x00000054
R1 0x00000065
R2 0x00000065
R3 0x40000000
R4 0x00000000
R5 0x00000000
R6 0x00000000
R7 0x00000000
R8 0x00000000
R9 0x00000000
R10 0x00000000
R11 0x00000000
R12 0x00000000
R13 (SP) 0x00000000
R14 (LR) 0x00000000
R15 (PC) 0x00000038
CPSR 0x60000063
SPSR 0x00000000
User/System
Fast Interrupt
Files
Symbols
01 ;/*Program to find largest of N numbers*/
02 AREA arr,code
03 ENTRY
04 main
05 ldr r0,=data1
06 ldr r3,=0x40000000
07 ldr r4,=0x05 ;//length of loop
08 ldr r1,[r0],#04
09 sub r4,r4,#01
10 back
11 ldr r2,[r0]
12 cmp r1,r2
13 bhs large ;// branch on low
14 mov r1,r2
15 large
16 add r0,r0,#04
17 sub r4,r4,#01
18 cmp r4,#00
19 bne back
20 str r1,[r3] ;// smallest value stored in memory location
21 stop b stop
22 AREA data,code
23 data1 dcd &64,&05,&9,&00,&65
24 END
larg.s
*** Restricted Version with 32768 Byte Code Size Limit
*** Currently used: 84 Bytes (0%)
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE DEFINE DIR Display Enter
Build
Command
Find in Files
Output Window
Filter: Execution-Mixed
Nr.
Address
Opcode
Instruction
Trace
Real-Time Agent: Target Stopped
Simulation
t1: 1.87227000 sec
L:16 C:28
NUM
R/W

```
        ;/*Program to convert Hex to ascii*/


        AREA arr,code
        entry
main
        ldr r0,=value1
        ldr r1,[r0]
        mov r2,r1
        cmp r1,#0x09
        bhi grt
        add r1,r1,#0x30                 ;//add 30 if data <9
        bl nxt1
grt
        add r1,r1,#0x37                 ;//add 37h if data>9
nxt1
        ldr r4,=0x40000000
        str r1,[r4]
stop b stop
        AREA data,code
value1 dcd &07
        END
```


HEX ASCII - µVision3 - [F:\Study Softwares\ARM\H2A.S]
File Edit View Project Debug Flash Peripherals Tools SVCS Window Help
Project Workspace
Register
Value
Current
R0 0x00000030
R1 0x00000037
R2 0x00000007
R3 0x00000000
R4 0x40000000
R5 0x00000000
R6 0x00000000
R7 0x00000000
R8 0x00000000
R9 0x00000000
R10 0x00000000
R11 0x00000000
R12 0x00000000
R13 (SP) 0x00000000
R14 (LR) 0x0000001c
R15 (PC) 0x00000028
CPSR 0x800000d3
SPSR 0x00000000
User/System
Fast Interrupt
Symbols
01 ;/*Program to convert Hex to ascii*/
02 TTL h2a
03 AREA arr,code
04 entry
05 main
06 ldr r0,=value1
07 ldr r1,[r0]
08 mov r2,r1
09 cmp r1,#0x09
10 bhi grt
11 add r1,r1,#0x30 ;//add 30 if data <9
12 bl nxt1
13 grt
14 add r1,r1,#0x37 ;//add 37h if data>9
15 nxt1
16 ldr r4,=0x40000000
17 str r1,[r4]
18 stop b stop
19 AREA data,code
20 value1 dcd &07
21 END
H2A.S
*** Restricted Version with 32768 Byte Code Size Limit
*** Currently used: 52 Bytes (0%)
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE DEFINE DIR Display Enter
Build Command Find in Files
Output Window
Filter: Execution-Mixed
Nr. Address Opcode Instruction
Trace
Real-Time Agent: Target Stopped
Simulation
t1: 3.17953483 sec
L:21 C:37
CAP NUM R/W
10:35 AM
4/1/2014

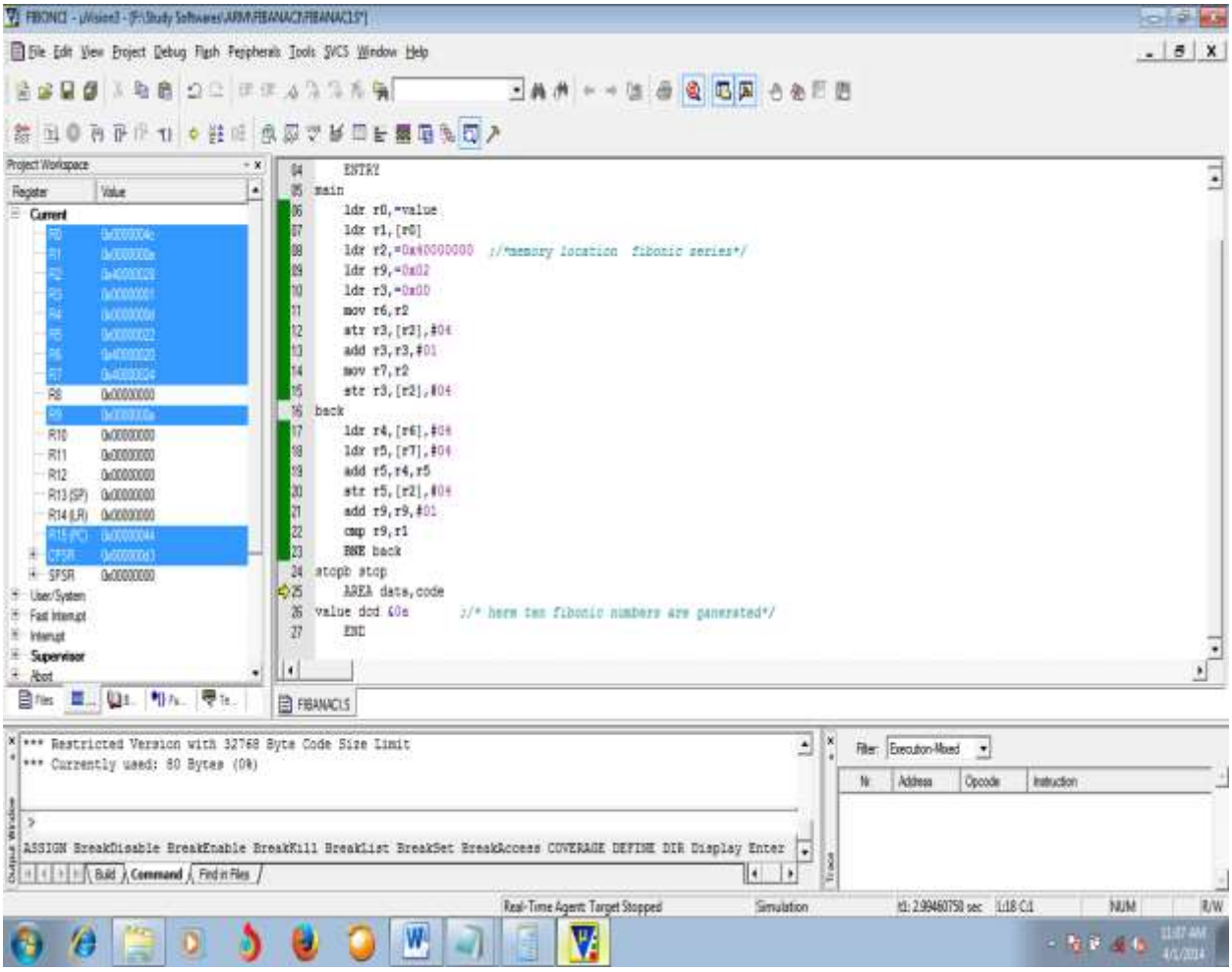
```
        ;/*Program to convert Ascii to hex*/


        AREA arr,code
        entry
main
        ldr r0,=value1
        ldr r1,[r0]
        mov r2,r1
        cmp r1,#0x39
        bhi grt
        sub r1,r1,#0x30                 ;//SUB 30 if data <39
        bl nxt1
grt
        sub r1,r1,#0x37                 ;//SUB 37h if data>39
nxt1
        ldr r4,=0x40000000
        str r1,[r4]
stop b stop
        AREA data,code
value1 dcd &41
        END
```


A2H - µVision3 - [F:\Study Softwares\ARM\ASCII 2HEX\A2H.S]
File Edit View Project Debug Flash Peripherals Tools SVCS Window Help
Project Workspace
Register
Value
Current
R0 0x00000030
R1 0x0000003a
R2 0x00000041
R3 0x00000000
R4 0x40000000
R5 0x00000000
R6 0x00000000
R7 0x00000000
R8 0x00000000
R9 0x00000000
R10 0x00000000
R11 0x00000000
R12 0x00000000
R13 (SP) 0x00000000
R14 (LR) 0x00000000
R15 (PC) 0x00000028
CPSR 0x200000d3
SPSR 0x00000000
User/System
Fast Interrupt
Symbols
01 ;/*Program to convert Ascii to hex*/
02 TTL h2a
03 AREA arr,code
04 entry
05 main
06 ldr r0,=value1
07 ldr r1,[r0]
08 mov r2,r1
09 cmp r1,#0x39
10 bhi grt
11 sub r1,r1,#0x30 ;//SUB 30 if data <39
12 bl nxt1
13 grt
14 sub r1,r1,#0x37 ;//SUB 37h if data>39
15 nxt1
16 ldr r4,=0x40000000
17 str r1,[r4]
18 stop b stop
19 AREA data,code
20 value1 dcd &41
21 END
A2H.S
*** Restricted Version with 32768 Byte Code Size Limit
*** Currently used: 52 Bytes (0%)
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE DEFINE DIR Display Enter
Build
Command
Find in Files
Filter: Execution-Mixed
Nr.
Address
Opcode
Instruction
Ready
Real-Time Agent: Target Stopped
Simulation
t1: 8.46749175 sec
L:18 C:1
CAP NUM
R/W

```
                ;/*Progarm to generate N Fibonic numbers*/


        AREA arr,code
        ENTRY
main
        ldr r0,=value
        ldr r1,[r0]
        ldr r2,=0x40000000      ;/*memory location fibonic series*/
        ldr r9,=0x02
        ldr r3,=0x00
        mov r6,r2
        str r3,[r2],#04
        add r3,r3,#01
        mov r7,r2
        str r3,[r2],#04
back
        ldr r4,[r6],#04
        ldr r5,[r7],#04
        add     r5,r4,r5
        str r5,[r2],#04
        add r9,r9,#01
        cmp r9,r1
        BNE back
stopb stop
        AREA data,code
```

value dcd &0a ;/* here ten fibonic numbers are ganerated*/

END



```
;//* To find the factorial of a given number using subroutine*//

        AREA arr,code
        ENTRY
main
    ldr r0,=value
        bl fact             ;// call subroutine fact
        ldr r1,=0x40000000
        str r5,[r1]
stop b stop
        AREA data,code
value dcd &0a
fact
  mov r6,r14
        ldr r2,[r0]
        cmp r2,#00
        beq END1
        mov r3,r2
loop
        sub r2,r2,#01
        cmp r2,#00
        mulne r3,r2,r3
        bne loop
        mov r5,r3
        bl END2
```

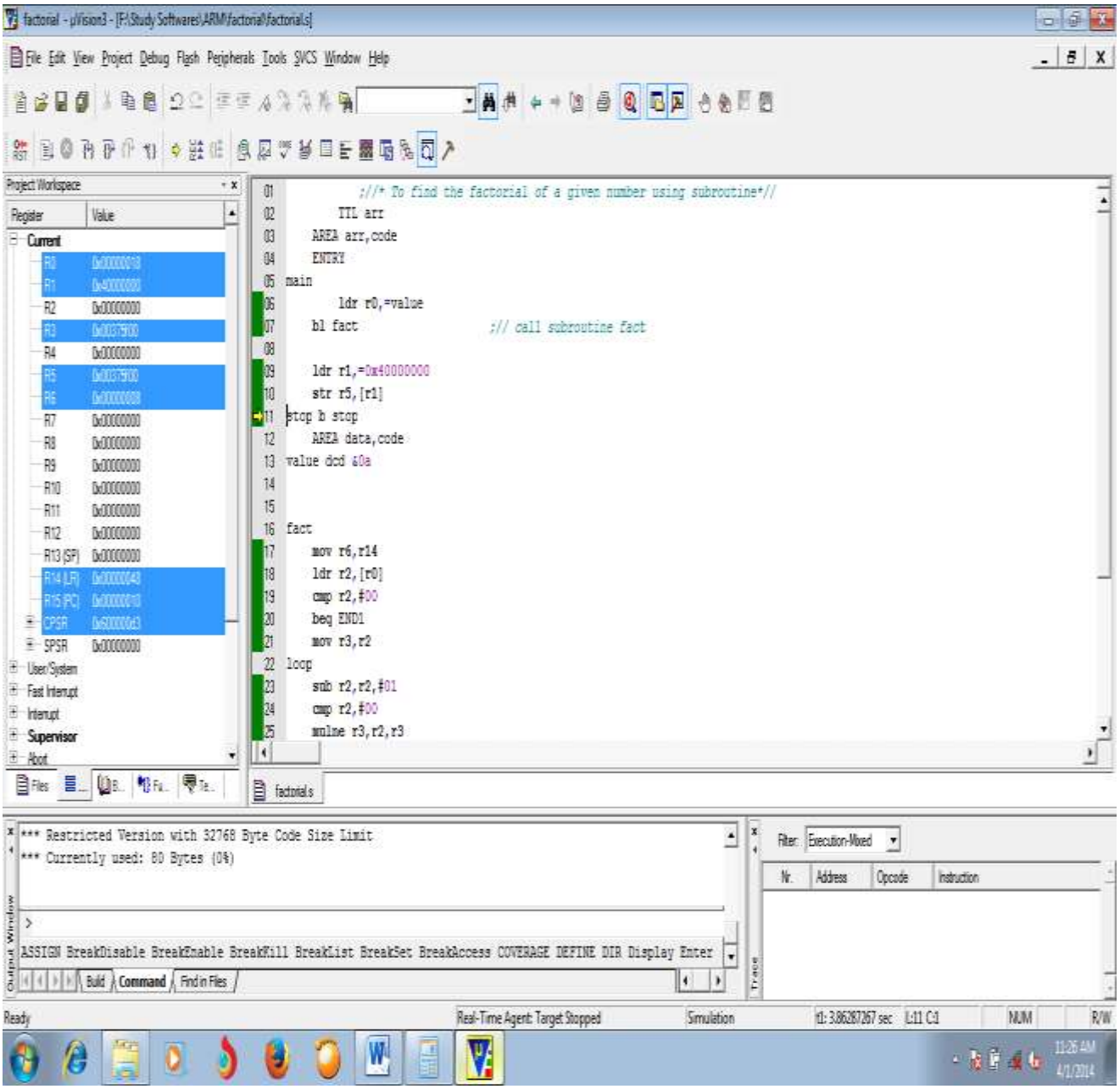
END1

ldr r5,=0x01

END2

mov PC,r6 ;// return to main program.

END



```
;//* To find the multiplication of two 32 bit number *//
AREA ADDTIN,CODE
        ENTRY
        ldr r0,value1
        ldr r1,value2
  umull r4,r3,r1,r0
value1 dcd &BBBBBBBB
value2 dcd &CCCCCCCC
        end
```