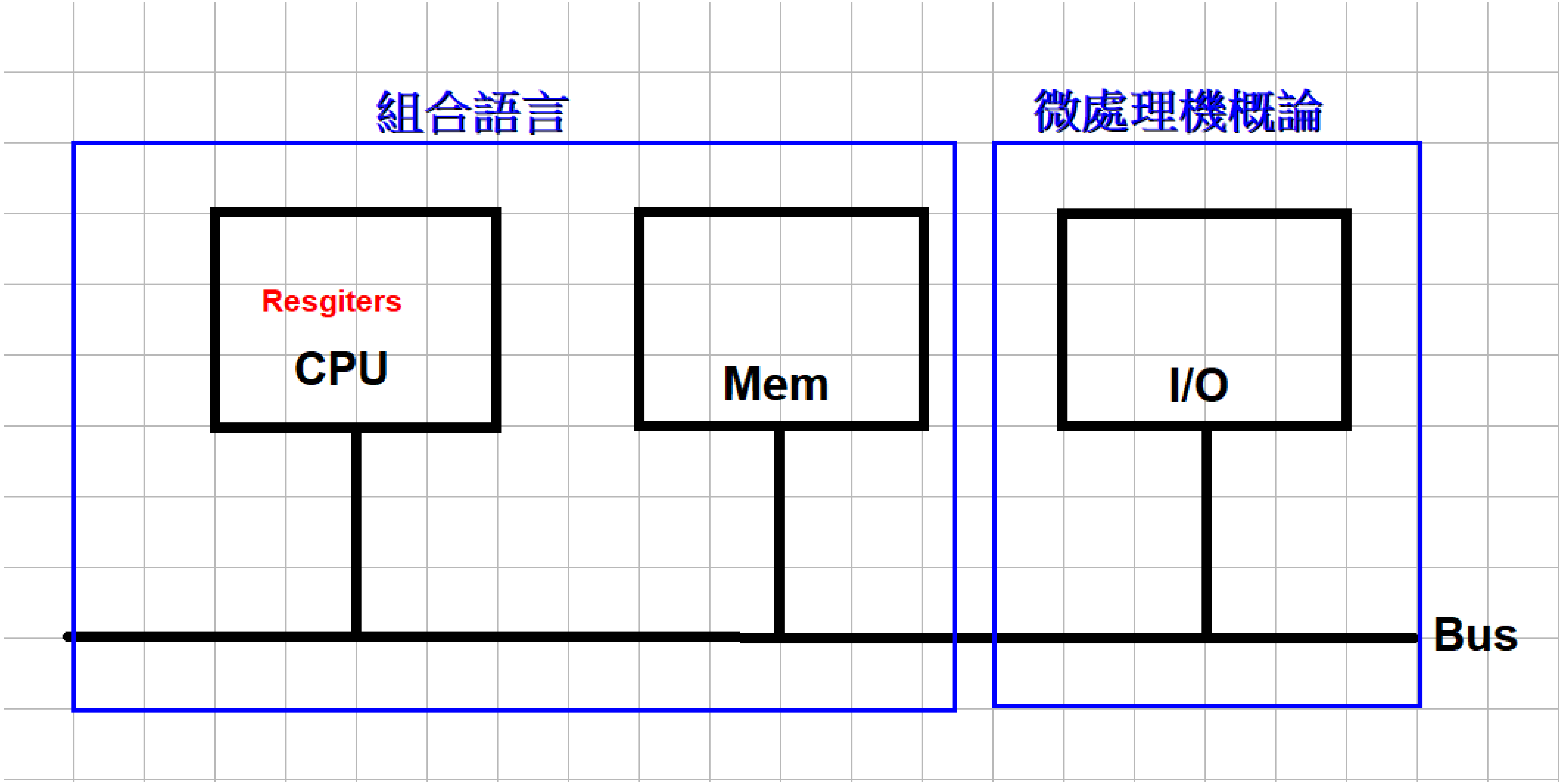
第 02 次組語實習課

學生 ： 林培瑋

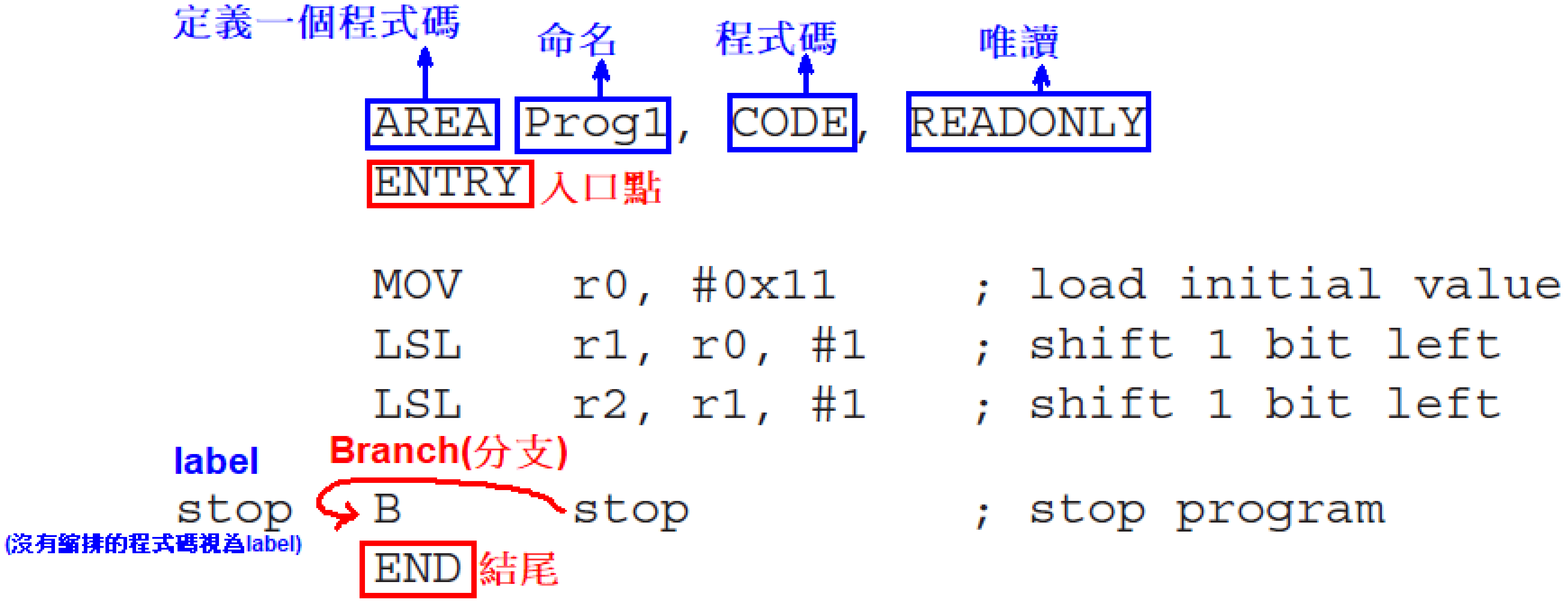
2023 Advanced Mixed-Operation System (AMOS) Lab.

Tamkang University
Department of Electrical and Computer Engineering
No.151, Yingzhuan Rd., Tamsui Dist., New Taipei City 25137, Taiwan (R.O.C.)

Introduction

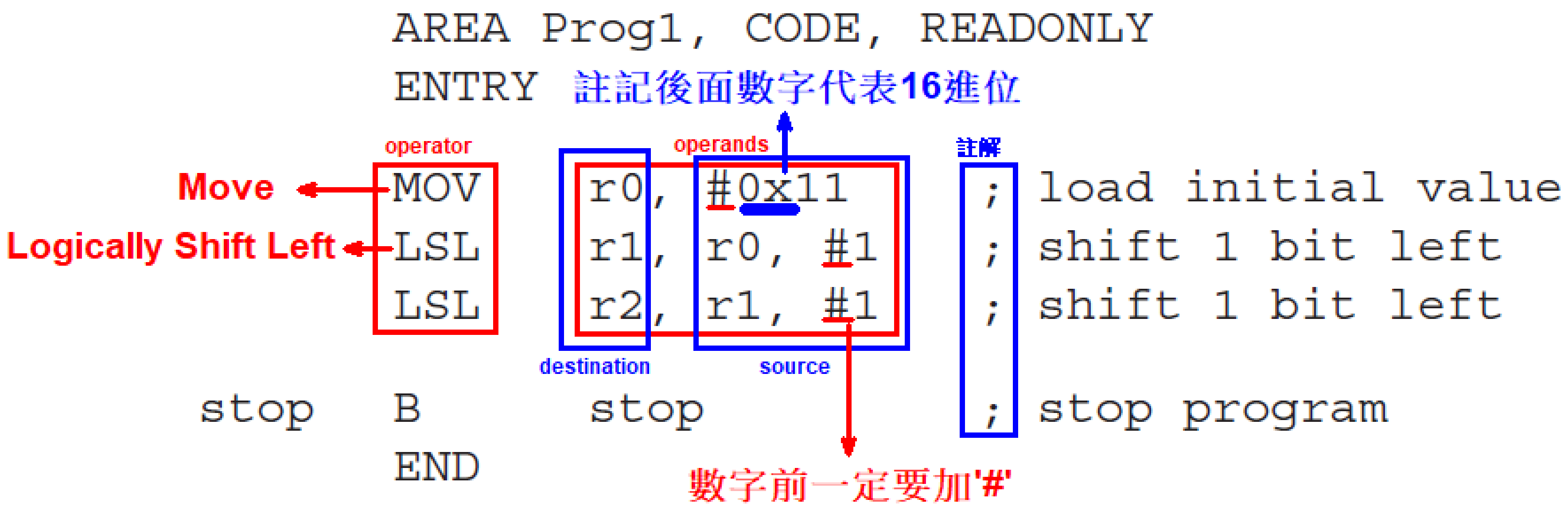


Program 1 ： Shifting Data(1/2)



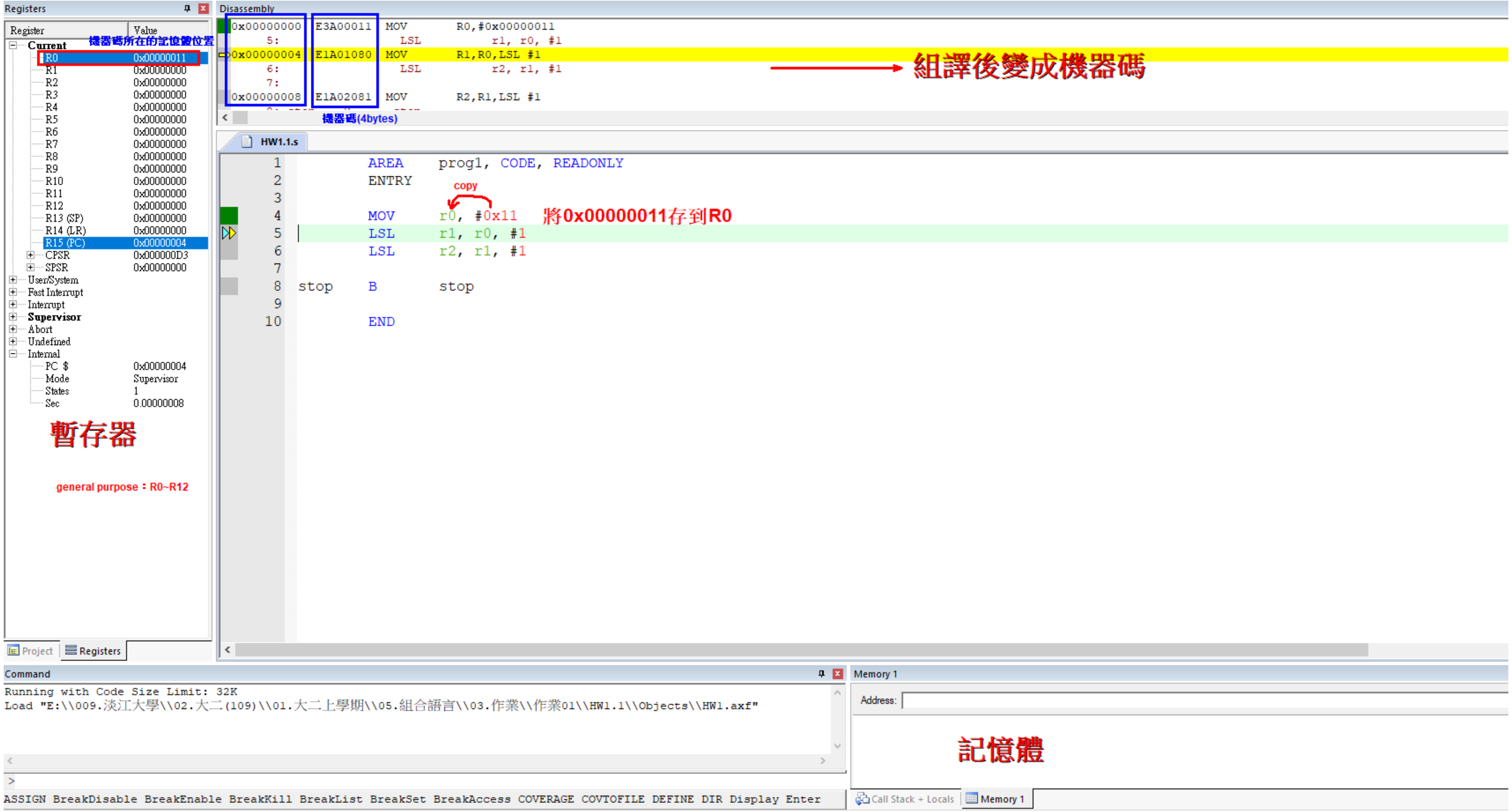
```
        AREA  Prog1, CODE, READONLY
        ENTRY

        MOV    r0, #0x11         ; load initial value
        LSL    r1, r0, #1        ; shift 1 bit left
        LSL    r2, r1, #1        ; shift 1 bit left

stop    B      stop              ; stop program
        END
```

Program 1 ： Shifting Data (2/2)



```
        AREA Prog1, CODE, READONLY
        ENTRY
        MOV   r0, #0x11     ; load initial value
        LSL   r1, r0, #1    ; shift 1 bit left
        LSL   r2, r1, #1    ; shift 1 bit left

stop    B     stop          ; stop program
        END
```

Program 1 ： Shifting Data using Keil Tool(1/3)



Registers

機器碼所在的記憶體位置

Register	Value
Current	
R0	0x00000011
R1	0x00000000
R2	0x00000000
R3	0x00000000
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)	0x00000004
CPSR	0x000000D3
SPSR	0x00000000
User/System	
Fast Interrupt	
Interrupt	
Supervisor	
Abort	
Undefined	
Internal	
PC $	0x00000004
Mode	Supervisor
States	1
Sec	0.00000008

暫存器

general purpose：R0~R12

Disassembly

```
0x00000000  E3A00011  MOV       R0,#0x00000011
     5:                 LSL           r1, r0, #1
0x00000004  E1A01080  MOV       R1,R0,LSL #1
     6:                 LSL           r2, r1, #1
     7:
0x00000008  E1A02081  MOV       R2,R1,LSL #1
```

機器碼(4bytes)

組譯後變成機器碼

HW1.1.s

```
        AREA    prog1, CODE, READONLY
        ENTRY

        MOV     r0, #0x11
        LSL     r1, r0, #1
        LSL     r2, r1, #1

stop    B       stop

        END
```

copy

將0x00000011存到R0

Command

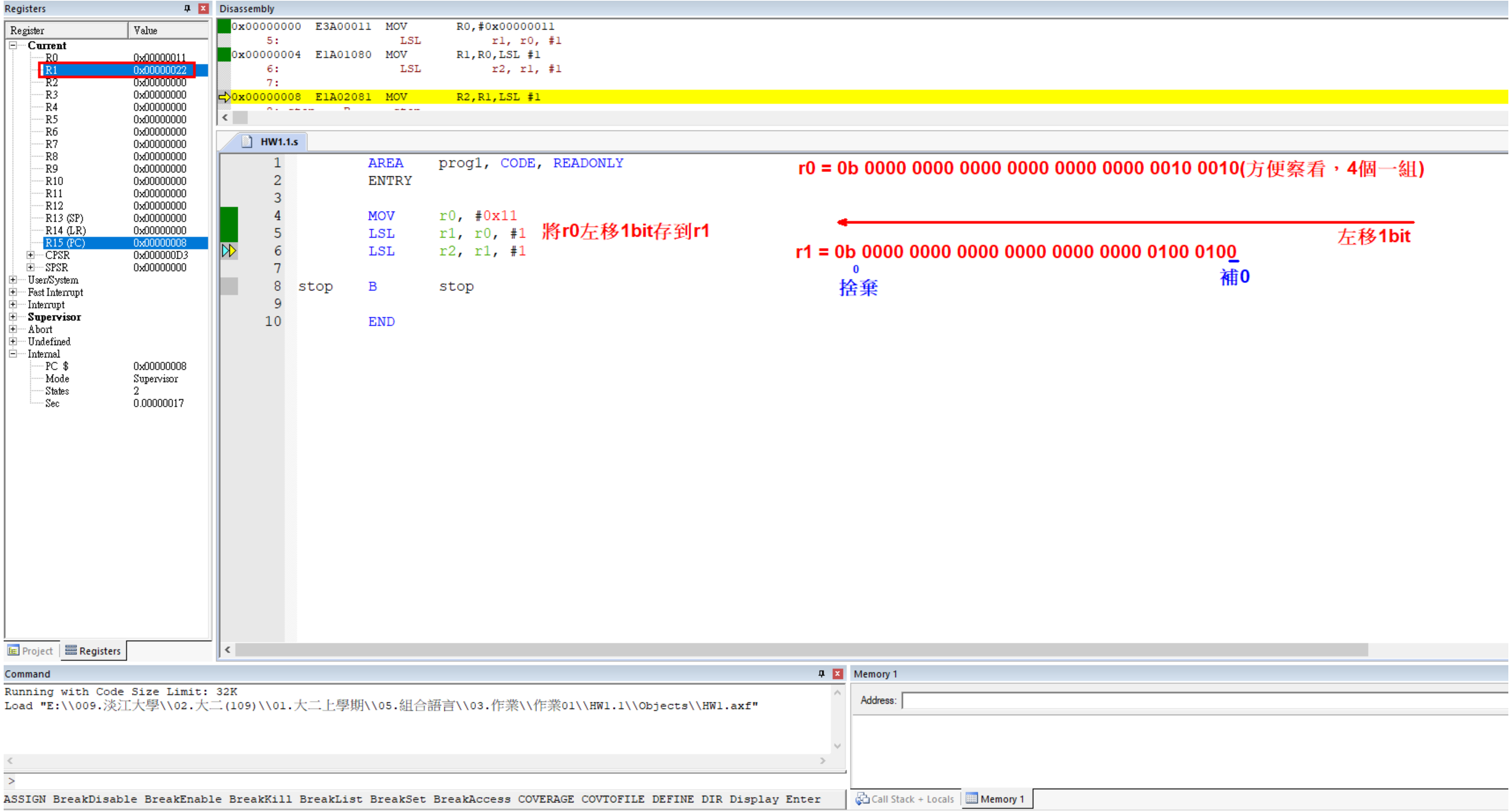
```
Running with Code Size Limit: 32K
Load "E:\\009.淡江大學\\02.大二(109)\\01.大二上學期\\05.組合語言\\03.作業\\作業01\\HW1.1\\Objects\\HW1.axf"

>
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE COVTOFILE DEFINE DIR Display Enter
```

Memory 1

Address:

記憶體

Program 1 ： Shifting Data using Keil Tool(2/3)



```
Register        Value
Current
  R0            0x00000011
  R1            0x00000022
  R2            0x00000000
  R3            0x00000000
  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)      0x00000008
  CPSR          0x000000D3
  SPSR          0x00000000
User/System
Fast Interrupt
Interrupt
Supervisor
Abort
Undefined
Internal
  PC $          0x00000008
  Mode          Supervisor
  States        2
  Sec           0.00000017
```

Disassembly

```
0x00000000  E3A00011  MOV       R0,#0x00000011
     5:                  LSL          r1, r0, #1
0x00000004  E1A01080  MOV       R1,R0,LSL #1
     6:                  LSL          r2, r1, #1
     7:
0x00000008  E1A02081  MOV       R2,R1,LSL #1
```

HW1.1.s

```
        AREA    prog1, CODE, READONLY
        ENTRY

        MOV     r0, #0x11
        LSL     r1, r0, #1
        LSL     r2, r1, #1

stop    B       stop

        END
```

將r0左移1bit存到r1

r0 = 0b 0000 0000 0000 0000 0000 0000 0010 0010(方便察看，4個一組)

左移1bit

r1 = 0b 0000 0000 0000 0000 0000 0000 0100 0100

0

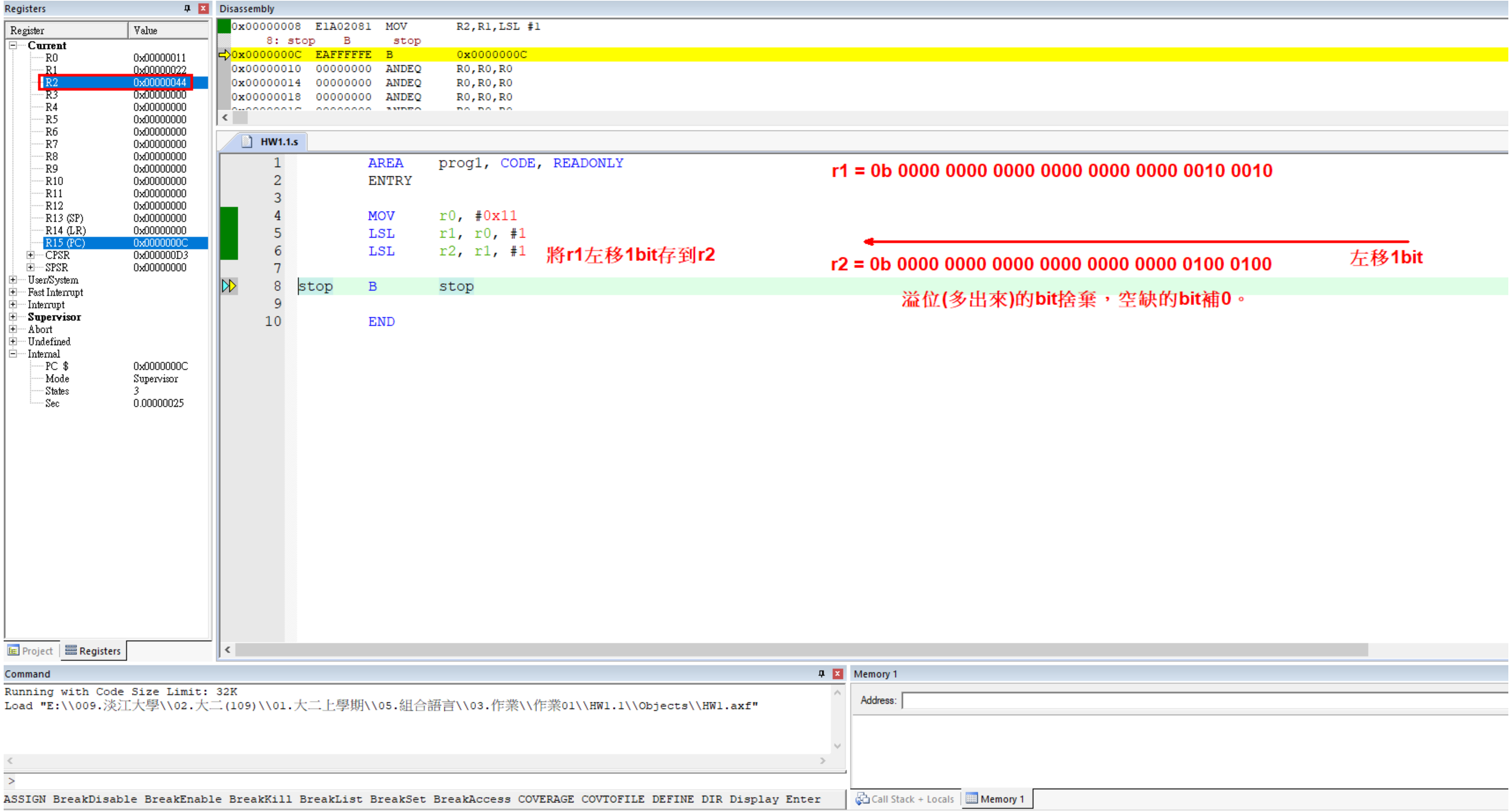
捨棄

補0

Command

```
Running with Code Size Limit: 32K
Load "E:\\009.淡江大學\\02.大二(109)\\01.大二上學期\\05.組合語言\\03.作業\\作業01\\HW1.1\\Objects\\HW1.axf"

>
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE COVTOFILE DEFINE DIR Display Enter
```

Program 1 ： Shifting Data using Keil Tool(3/3)



```
Registers
Register        Value
Current
  R0            0x00000011
  R1            0x00000022
  R2            0x00000044
  R3            0x00000000
  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)      0x0000000C
  CPSR          0x000000D3
  SPSR          0x00000000
User/System
Fast Interrupt
Interrupt
Supervisor
Abort
Undefined
Internal
  PC $          0x0000000C
  Mode          Supervisor
  States        3
  Sec           0.00000025
```

```
Disassembly
0x00000008  E1A02081  MOV       R2,R1,LSL #1
     8: stop     B        stop
0x0000000C  EAFFFFFE  B         0x0000000C
0x00000010  00000000  ANDEQ     R0,R0,R0
0x00000014  00000000  ANDEQ     R0,R0,R0
0x00000018  00000000  ANDEQ     R0,R0,R0
```

HW1.1.s

```
          AREA     prog1, CODE, READONLY
          ENTRY

          MOV      r0, #0x11
          LSL      r1, r0, #1
          LSL      r2, r1, #1

stop      B        stop

          END
```

將r1左移1bit存到r2

r1 = 0b 0000 0000 0000 0000 0000 0000 0010 0010

左移1bit

r2 = 0b 0000 0000 0000 0000 0000 0000 0100 0100

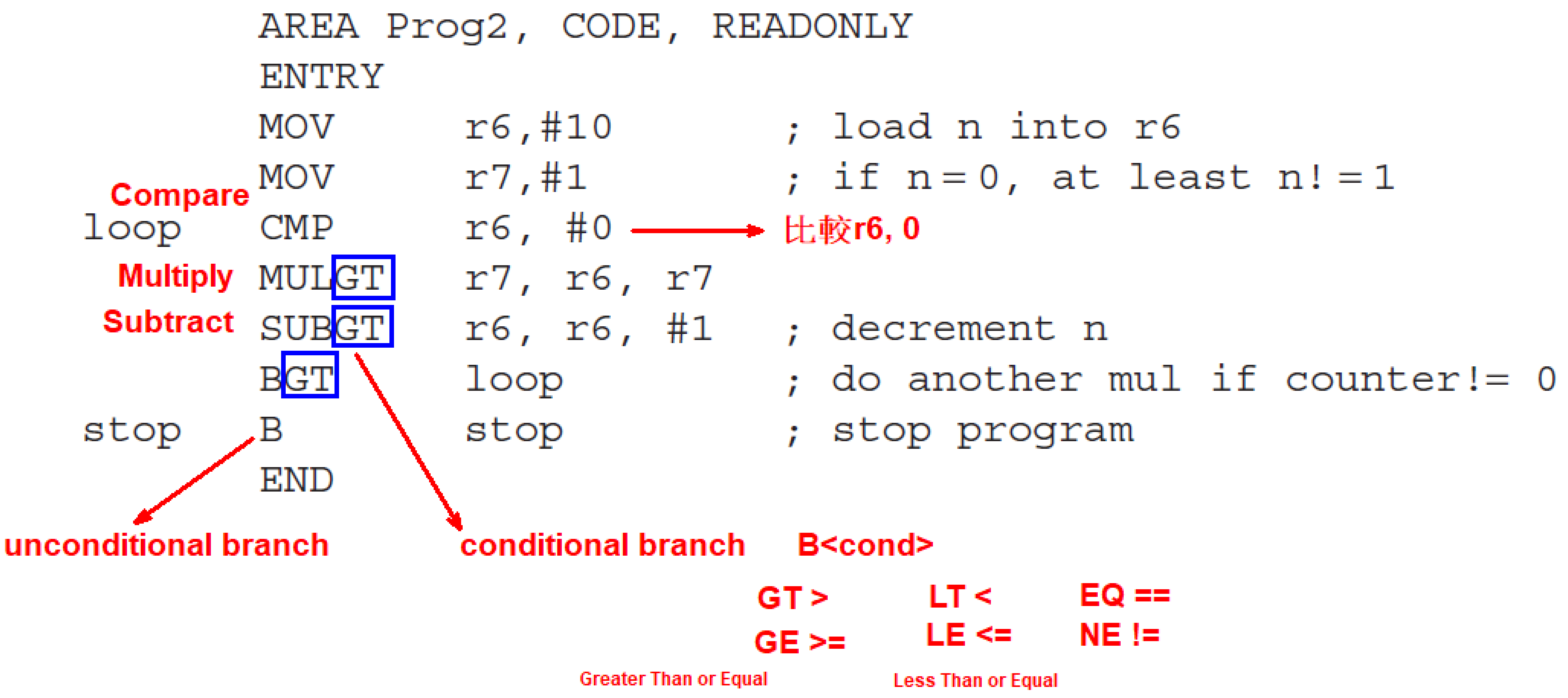
溢位(多出來)的bit捨棄，空缺的bit補0。

```
Command
Running with Code Size Limit: 32K
Load "E:\\009.淡江大學\\02.大二(109)\\01.大二上學期\\05.組合語言\\03.作業\\作業01\\HW1.1\\Objects\\HW1.axf"
>
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE COVTOFILE DEFINE DIR Display Enter
```

Memory 1

Address:

Program 2 ： Factorial Calculation



```
        AREA Prog2, CODE, READONLY
        ENTRY
        MOV     r6,#10          ; load n into r6
        MOV     r7,#1           ; if n=0, at least n!=1
loop    CMP     r6, #0
        MULGT   r7, r6, r7
        SUBGT   r6, r6, #1      ; decrement n
        BGT     loop            ; do another mul if counter!= 0
stop    B       stop            ; stop program
        END
```

Compare

比較r6, 0

Multiply

Subtract

unconditional branch

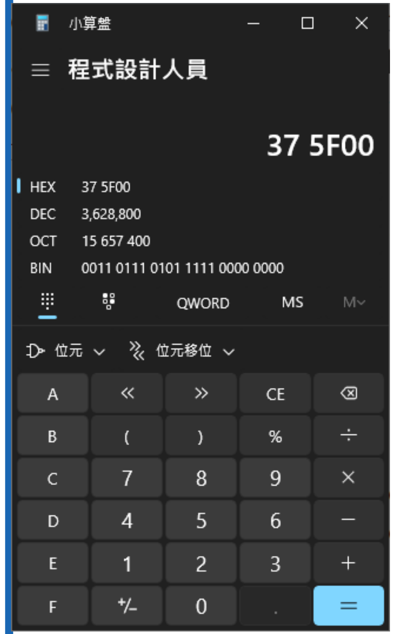
conditional branch B<cond>

GT >	LT <	EQ ==
GE >=	LE <=	NE !=

Greater Than or Equal

Less Than or Equal

Program 2 ： Factorial Calculation using Keil Tool

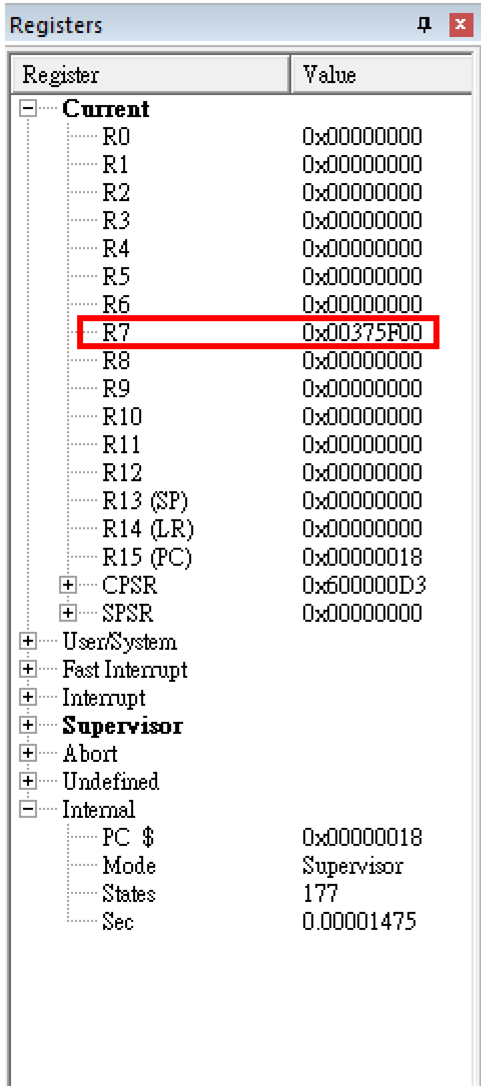


小算盤

程式設計人員

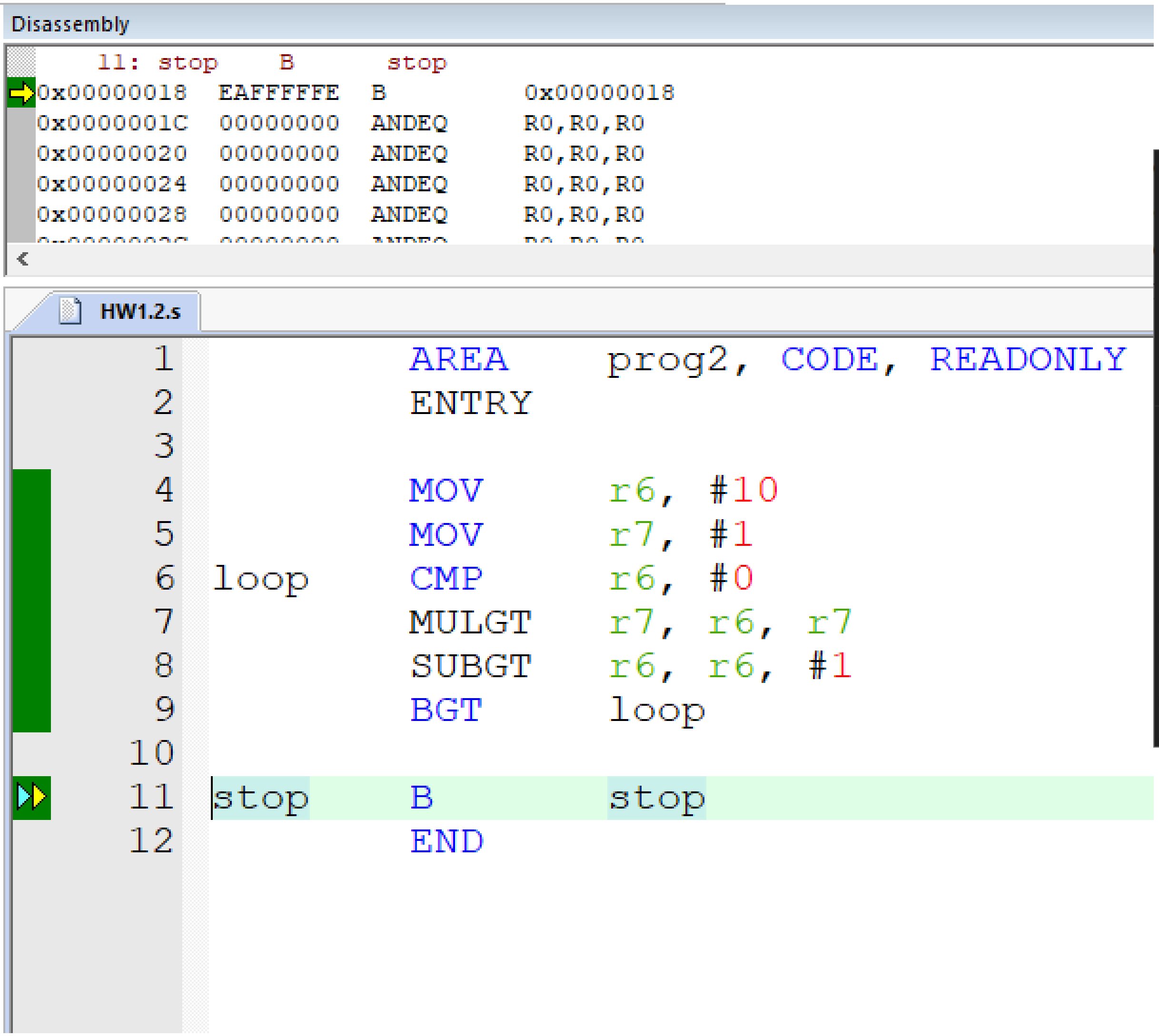
37 5F00

HEX 37 5F00
DEC 3,628,800
OCT 15 657 400
BIN 0011 0111 0101 1111 0000 0000



Registers

Register	Value
Current	
R0	0x00000000
R1	0x00000000
R2	0x00000000
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00375F00
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x00000018
CPSR	0x600000D3
SPSR	0x00000000
User/System	
Fast Interrupt	
Interrupt	
Supervisor	
Abort	
Undefined	
Internal	
PC $	0x00000018
Mode	Supervisor
States	177
Sec	0.00001475

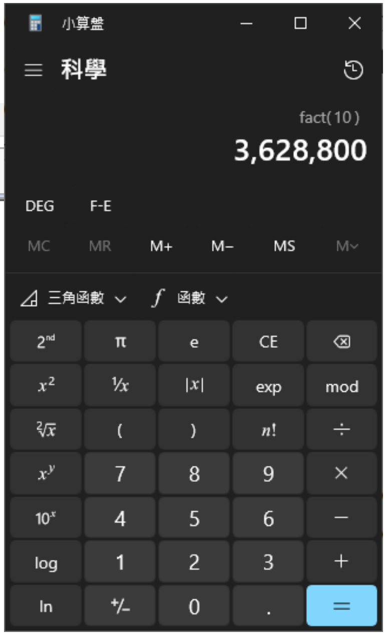


Disassembly

```
    11: stop     B       stop
0x00000018  EAFFFFFE  B       0x00000018
0x0000001C  00000000  ANDEQ   R0,R0,R0
0x00000020  00000000  ANDEQ   R0,R0,R0
0x00000024  00000000  ANDEQ   R0,R0,R0
0x00000028  00000000  ANDEQ   R0,R0,R0
```

HW1.2.s

```
        AREA    prog2, CODE, READONLY
        ENTRY

        MOV     r6, #10
        MOV     r7, #1
loop    CMP     r6, #0
        MULGT   r7, r6, r7
        SUBGT   r6, r6, #1
        BGT     loop

stop    B       stop
        END
```



小算盤

科學

fact(10)

3,628,800

Q&A

Thanks for your attention !!