EXPERIMENT 2

Aim:

Write a program in ARM Assembly language to load any register with 32 bit data and perform following

a. Shift left by 2 bits.
b. Shift right by number of bits stored in register.
c. Shift left 5 bits conditionally when ‘0’ flag is set.
d. Arithmetic shift by the value contained in register.

Tool used: Keil uVision4

Theory:

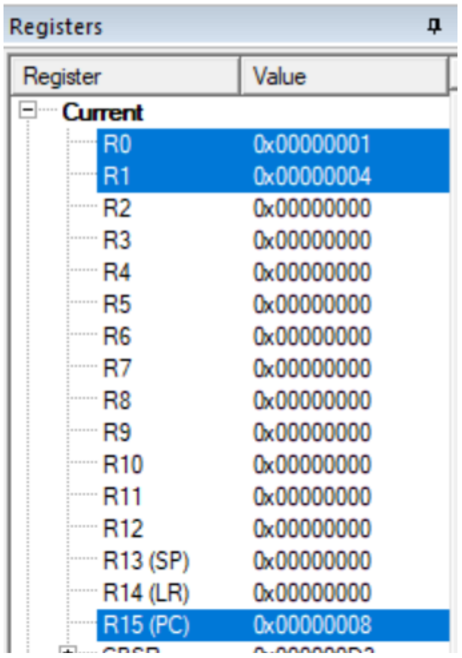
Here I have used LSL, LSR and ASR for shifting the bits. LSL is used for shifting the bits left and concatenate a 0 at the LSB. LSR is used for shifting the bits right and concatenate a 0 at the MSB. ASR is used to shift the bits right and concatenate the value of MSB at the new MSB

a) Shift left by 2 bits.

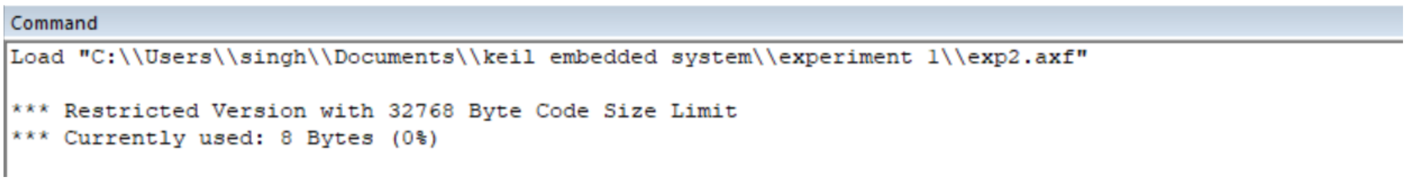
Code:

```
 AREA PROGRAM, CODE, READONLY
 ENTRY
MAIN
 LDR R0,=0x00000001 ;LOAD R0 with a value
 MOV R1,R0,LSL#0x02 ;move value of R0 to R1 with left shift by 2 bits
 END
```

Register Output:



```
Registers
Register        Value
Current
  R0            0x00000001
  R1            0x00000004
  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
```

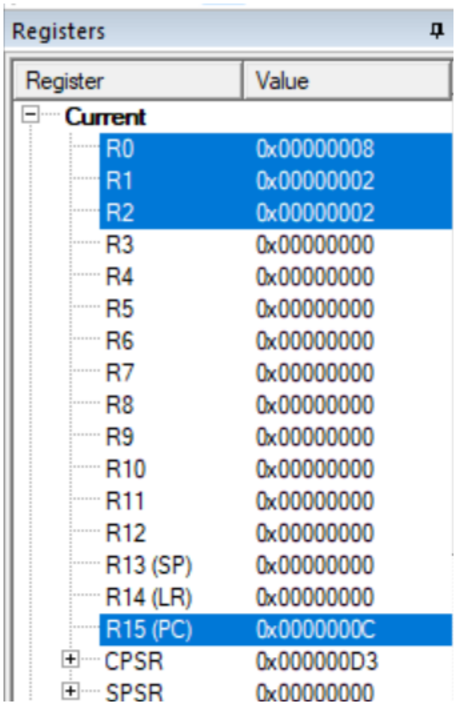

```
Command
Load "C:\\Users\\singh\\Documents\\keil embedded system\\experiment 1\\exp2.axf"

*** Restricted Version with 32768 Byte Code Size Limit
*** Currently used: 8 Bytes (0%)
```

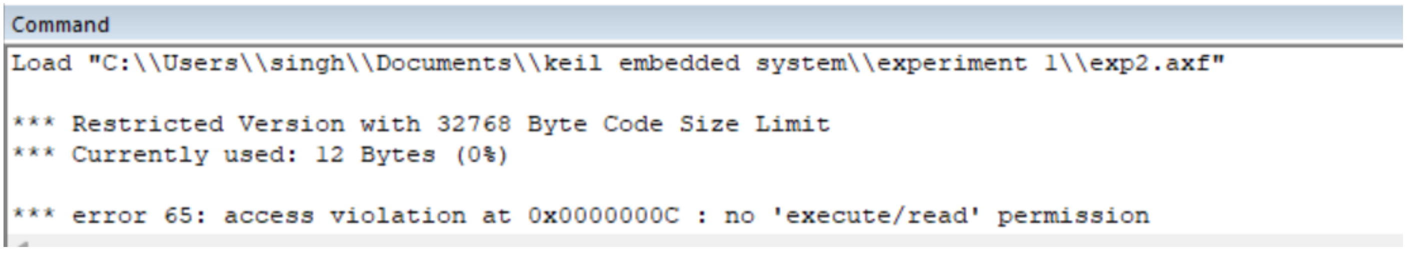
b) Shift right by number of bits stored in register.

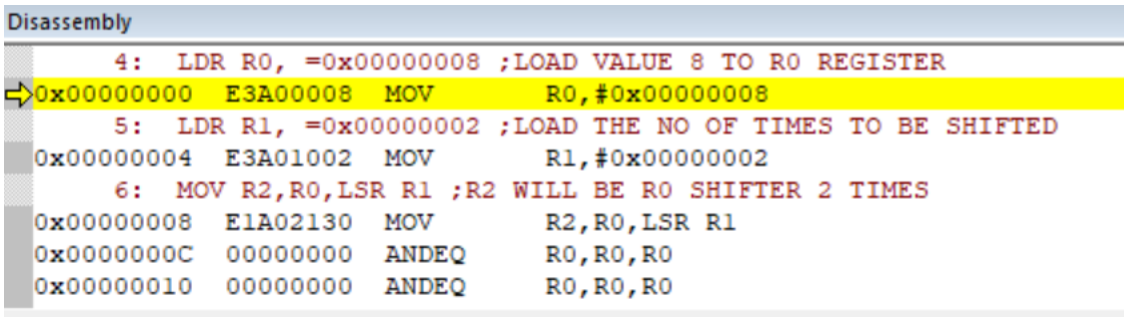
Code:

```
 AREA PROGRAM, CODE, READONLY
 ENTRY
MAIN
 LDR R0, =0x00000008 ;LOAD VALUE 8 TO R0 REGISTER
 LDR R1, =0x00000002 ;LOAD THE NO OF TIMES TO BE SHIFTED
 MOV R2,R0,LSR R1 ;R2 WILL BE R0 SHIFTER 2 TIMES
 END
```

Output:


```
Registers
Register        Value
Current
  R0            0x00000008
  R1            0x00000002
  R2            0x00000002
  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
```


```
Command
Load "C:\\Users\\singh\\Documents\\keil embedded system\\experiment 1\\exp2.axf"

*** Restricted Version with 32768 Byte Code Size Limit
*** Currently used: 12 Bytes (0%)

*** error 65: access violation at 0x0000000C : no 'execute/read' permission
```



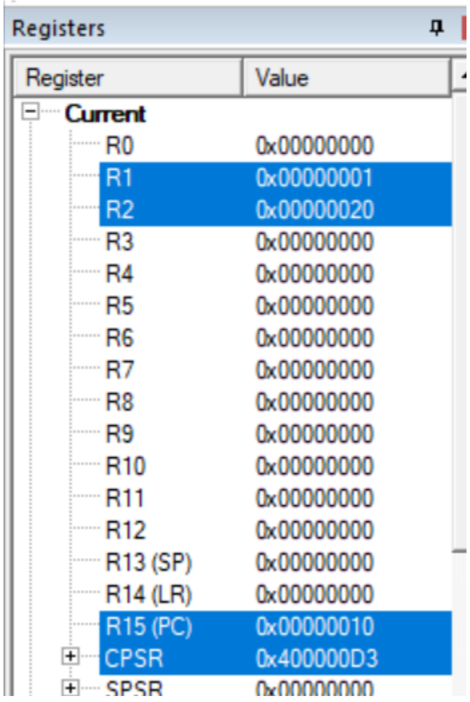
```
Disassembly
      4:  LDR R0, =0x00000008 ;LOAD VALUE 8 TO R0 REGISTER
0x00000000  E3A00008  MOV       R0,#0x00000008
      5:  LDR R1, =0x00000002 ;LOAD THE NO OF TIMES TO BE SHIFTED
0x00000004  E3A01002  MOV       R1,#0x00000002
      6:  MOV R2,R0,LSR R1 ;R2 WILL BE R0 SHIFTER 2 TIMES
0x00000008  E1A02130  MOV       R2,R0,LSR R1
0x0000000C  00000000  ANDEQ     R0,R0,R0
0x00000010  00000000  ANDEQ     R0,R0,R0
```

c) Shift left 5 bits conditionally when ‘0’ flag is set.

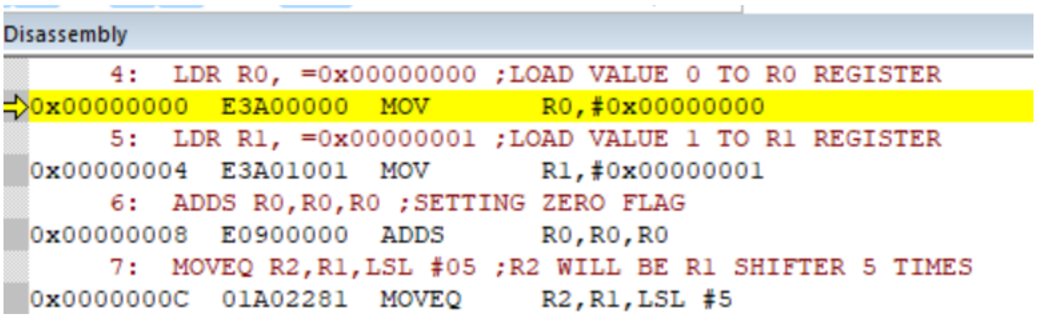
Code:

```
 AREA PROGRAM, CODE, READONLY
 ENTRY
MAIN
 LDR R0, =0x00000000 ;LOAD VALUE 0 TO R0 REGISTER
 LDR R1, =0x00000001 ;LOAD VALUE 1 TO R1 REGISTER
 ADDS R0,R0,R0 ;SETTING ZERO FLAG
 MOVEQ R2,R1,LSL #05 ;R2 WILL BE R1 SHIFTER 5 TIMES
 END
```

Output:



```
Registers
Register      Value
Current
  R0          0x00000000
  R1          0x00000001
  R2          0x00000020
  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)    0x00000010
  CPSR        0x400000D3
  SPSR        0x00000000
```



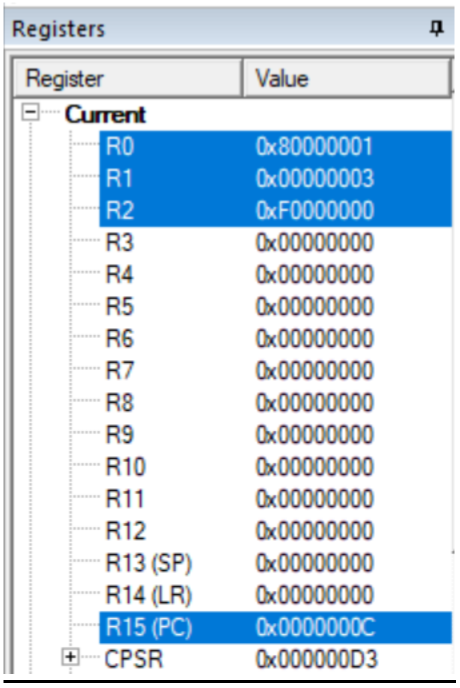
```
Disassembly
      4:  LDR R0, =0x00000000 ;LOAD VALUE 0 TO R0 REGISTER
0x00000000  E3A00000  MOV       R0,#0x00000000
      5:  LDR R1, =0x00000001 ;LOAD VALUE 1 TO R1 REGISTER
0x00000004  E3A01001  MOV       R1,#0x00000001
      6:  ADDS R0,R0,R0 ;SETTING ZERO FLAG
0x00000008  E0900000  ADDS      R0,R0,R0
      7:  MOVEQ R2,R1,LSL #05 ;R2 WILL BE R1 SHIFTER 5 TIMES
0x0000000C  01A02281  MOVEQ     R2,R1,LSL #5
```

d) Arithmetic Shift right by number of bits stored in register.

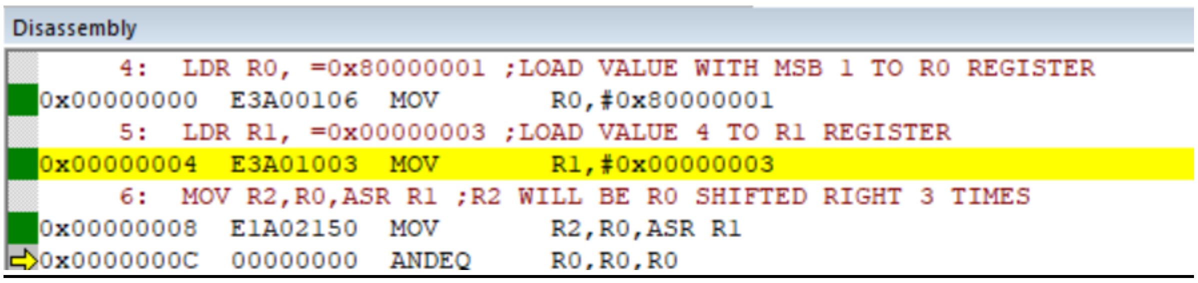
Code:

```
 AREA PROGRAM, CODE, READONLY
 ENTRY
MAIN
 LDR R0, =0x80000001 ;LOAD VALUE WITH MSB 1 TO R0 REGISTER
 LDR R1, =0x00000003 ;LOAD VALUE 3 TO R1 REGISTER
 MOV R2,R0,ASR R1 ;R2 WILL BE R0 SHIFTED RIGHT 3 TIMES
 END
```

Output:



```
Registers
Register        Value
Current
  R0            0x80000001
  R1            0x00000003
  R2            0xF0000000
  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
```



```
Disassembly
     4:  LDR R0, =0x80000001 ;LOAD VALUE WITH MSB 1 TO R0 REGISTER
0x00000000  E3A00106  MOV       R0,#0x80000001
     5:  LDR R1, =0x00000003 ;LOAD VALUE 4 TO R1 REGISTER
0x00000004  E3A01003  MOV       R1,#0x00000003
     6:  MOV R2,R0,ASR R1 ;R2 WILL BE R0 SHIFTED RIGHT 3 TIMES
0x00000008  E1A02150  MOV       R2,R0,ASR R1
0x0000000C  00000000  ANDEQ     R0,R0,R0
```

Result:

All the parts of the experiments are performed successfully and their results are also verified correctly.