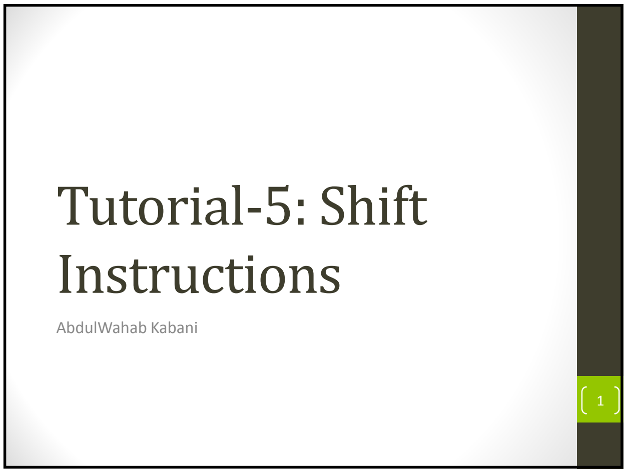
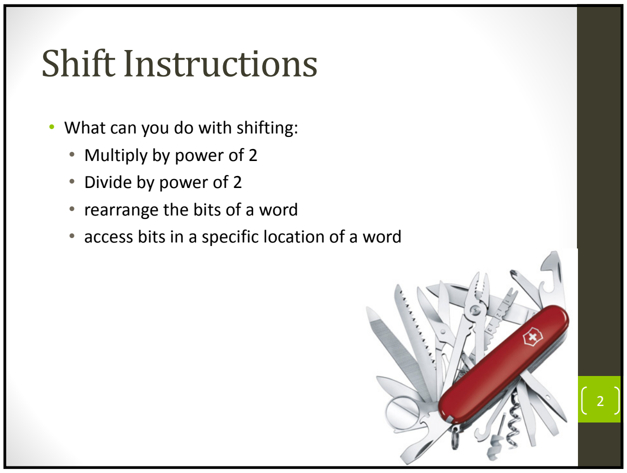


Tutorial-5: Shift Instructions

AbdulWahab Kabani

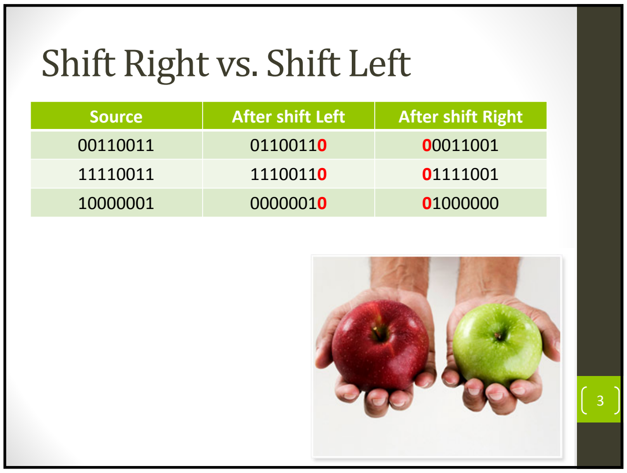
1



Shift Instructions

- What can you do with shifting:
  - Multiply by power of 2
  - Divide by power of 2
  - rearrange the bits of a word
  - access bits in a specific location of a word

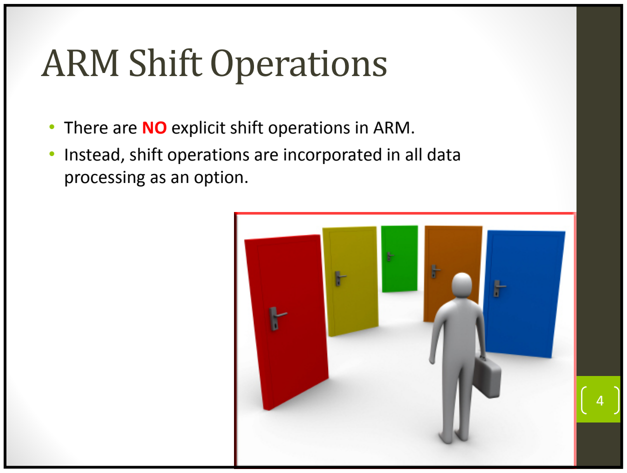
2



Shift Right vs. Shift Left

Source	After shift Left	After shift Right
00110011	01100110	00011001
11110011	11100110	01111001
10000001	00000010	01000000

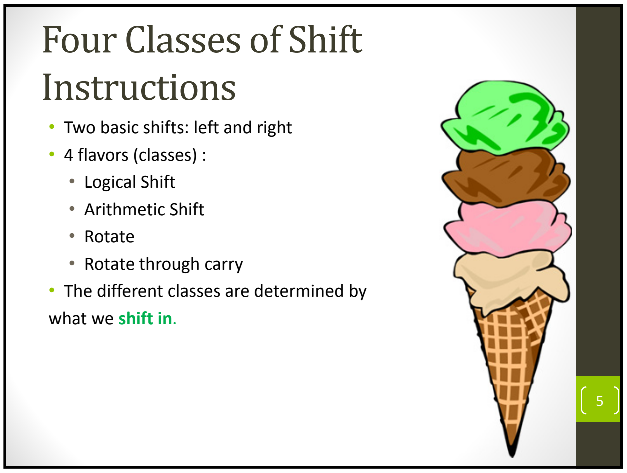
3



ARM Shift Operations

- There are **NO** explicit shift operations in ARM.
- Instead, shift operations are incorporated in all data processing as an option.

4

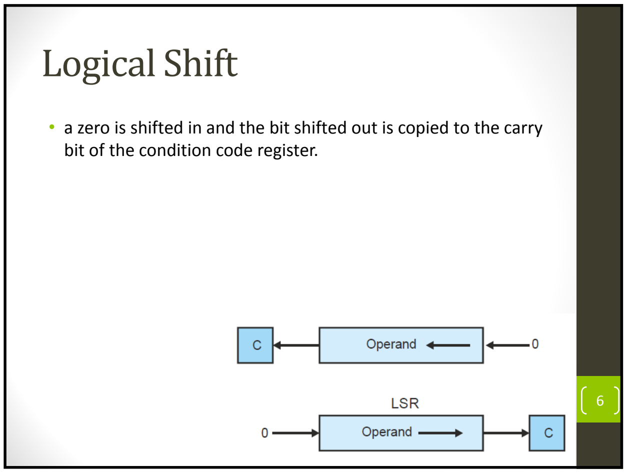


Four Classes of Shift Instructions

- Two basic shifts: left and right
- 4 flavors (classes) :
  - Logical Shift
  - Arithmetic Shift
  - Rotate
  - Rotate through carry
- The different classes are determined by

what we **shift in.**

5



Logical Shift

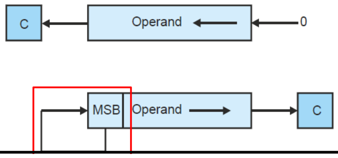
- a zero is shifted in and the bit shifted out is copied to the carry bit of the condition code register.

C
Operand
0
LSR
0
Operand
C

6

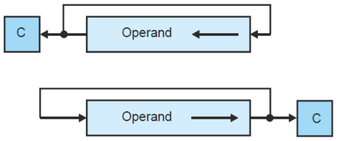
Arithmetic Shift

- The sign of a two's complement number is preserved. How?
- 11001010_2 = -54 (in two's complement)
- After Shift Right:
  - 11001010_2 (i.e., -54) → 11100101_2 (i.e., -27)



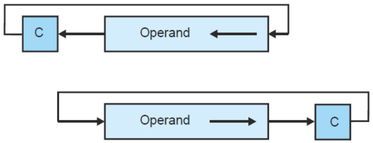
Rotate Shift

- the bit shifted out is copied into the bit vacated at the other end
- no bit is lost during a rotate



Rotate Through Carry

- Similar to Rotate Shift
- The carry bit is treated as part of the word to be shifted



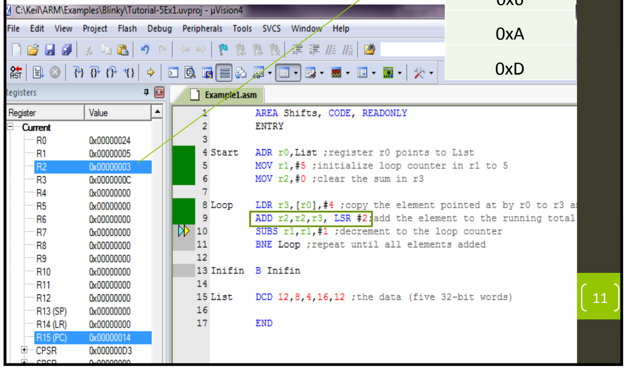
Example-1: Logical Shift

- We have a list of unsigned numbers.
- **Divide** them by 4 in an **efficient way** and sum them up.
- Example:

Number		Number/4	CumSum	CumSum (hex)
12	4	3	3	0x3
8	4	2	5	0x5
4	4	1	6	0x6
16	4	4	10	0xA
12	4	3	13	0xD
Sum		13		0xD

Example-1: Logical Shift

0x3
0x5
0x6
0xA
0xD

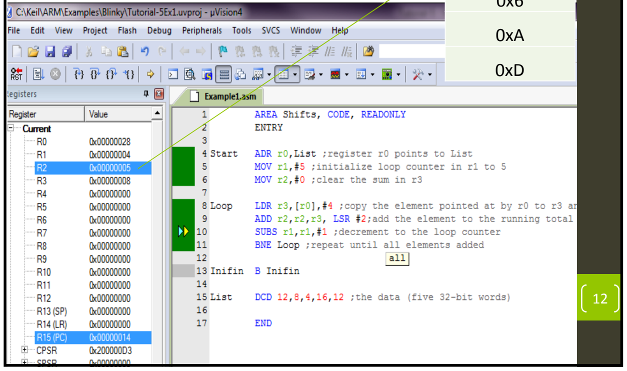


```
Register      Value
Current
  R0          0x00000024
  R1          0x00000005
  R2          0x00000003
  R3          0x0000000C
  R4          0x00000000
  R5          0x00000000
  R6          0x00000000
  R7          0x00000000
  R8          0x00000000
  R9          0x00000000
  R10         0x00000000
  R11         0x00000000
  R12         0x00000000
  R13 (SP)    0x00000000
  R14 (LR)    0x00000000
  R15 (PC)    0x00000014
  CPSR        0x000000D3
```

```
        AREA Shifts, CODE, READONLY
        ENTRY

Start   ADR r0,List ;register r0 points to List
        MOV r1,#5 ;initialize loop counter in r1 to 5
        MOV r2,#0 ;clear the sum in r3

Loop    LDR r3,[r0],#4 ;copy the element pointed at by r0 to r3 a
        ADD r2,r2,r3, LSR #2;add the element to the running total
        SUBS r1,r1,#1 ;decrement to the loop counter
        BNE Loop ;repeat until all elements added

Inifin  B Inifin

List    DCD 12,8,4,16,12 ;the data (five 32-bit words)

        END
```

Example-1: Logical Shift

0x3
0x5
0x6
0xA
0xD

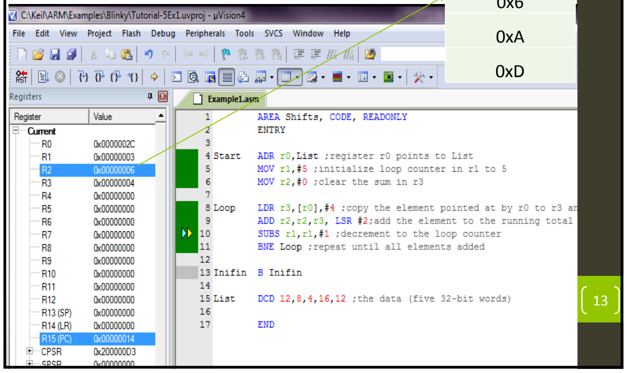


```
Register      Value
Current
  R0          0x00000028
  R1          0x00000004
  R2          0x00000005
  R3          0x00000008
  R4          0x00000000
  R5          0x00000000
  R6          0x00000000
  R7          0x00000000
  R8          0x00000000
  R9          0x00000000
  R10         0x00000000
  R11         0x00000000
  R12         0x00000000
  R13 (SP)    0x00000000
  R14 (LR)    0x00000000
  R15 (PC)    0x00000014
  CPSR        0x200000D3
```

```
        AREA Shifts, CODE, READONLY
        ENTRY

Start   ADR r0,List ;register r0 points to List
        MOV r1,#5 ;initialize loop counter in r1 to 5
        MOV r2,#0 ;clear the sum in r3

Loop    LDR r3,[r0],#4 ;copy the element pointed at by r0 to r3 ar
        ADD r2,r2,r3, LSR #2;add the element to the running total
        SUBS r1,r1,#1 ;decrement to the loop counter
        BNE Loop ;repeat until all elements added

Inifin  B Inifin

List    DCD 12,8,4,16,12 ;the data (five 32-bit words)

        END
```

Example-1: Logical Shift

0x3
0x5
0x6
0xA
0xD


C:\Keil\ARM\Examples\Blinky\Tutorial-5Ex1.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Example1.asm

```
Register      Value
Current
  R0          0x0000002C
  R1          0x00000003
  R2          0x00000006
  R3          0x00000004
  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)    0x00000014
  CPSR        0x200000D3
```

```
        AREA Shifts, CODE, READONLY
        ENTRY

Start   ADR r0,List ;register r0 points to List
        MOV r1,#5 ;initialize loop counter in r1 to 5
        MOV r2,#0 ;clear the sum in r3

Loop    LDR r3,[r0],#4 ;copy the element pointed at by r0 to r3 an
        ADD r2,r2,r3, LSR #2;add the element to the running total
        SUBS r1,r1,#1 ;decrement to the loop counter
        BNE Loop ;repeat until all elements added

Inifin  B Inifin

List    DCD 12,8,4,16,12 ;the data (five 32-bit words)

        END
```

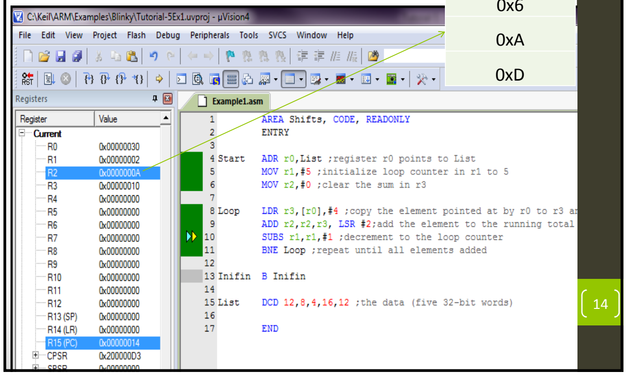
Example-2: Logical Shift

- We have a list of unsigned numbers.
- **Multiply** them by 4 in an **efficient way** and sum them up.
- Example:

Number		NumberX4	CumSum	CumSum (hex)
12	4	48	48	0x30
8	4	32	80	0x50
4	4	16	96	0x60
16	4	64	160	0xA0
12	4	48	208	0xD0
Sum		**208**		**0xD0**

Example-1: Logical Shift

0x3
0x5
0x6
0xA
0xD

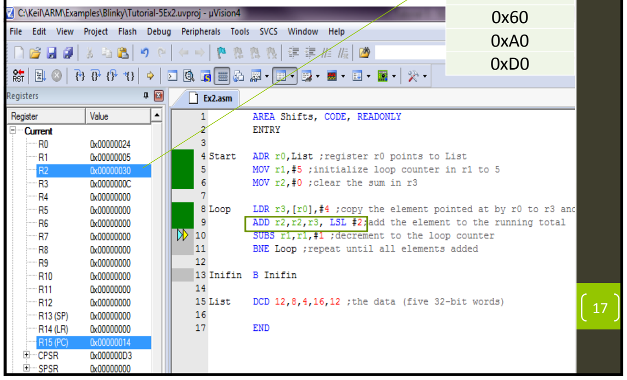

C:\Keil\ARM\Examples\Blinky\Tutorial-5Ex1.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Example1.asm

```
Register      Value
Current
  R0          0x00000030
  R1          0x00000002
  R2          0x0000000A
  R3          0x00000010
  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)    0x00000014
  CPSR        0x200000D3
```

```
        AREA Shifts, CODE, READONLY
        ENTRY

Start   ADR r0,List ;register r0 points to List
        MOV r1,#5 ;initialize loop counter in r1 to 5
        MOV r2,#0 ;clear the sum in r3

Loop    LDR r3,[r0],#4 ;copy the element pointed at by r0 to r3 ar
        ADD r2,r2,r3, LSR #2;add the element to the running total
        SUBS r1,r1,#1 ;decrement to the loop counter
        BNE Loop ;repeat until all elements added

Inifin  B Inifin

List    DCD 12,8,4,16,12 ;the data (five 32-bit words)

        END
```

Example-2: Logical Shift

CumSum (hex)
0x30
0x50
0x60
0xA0
0xD0

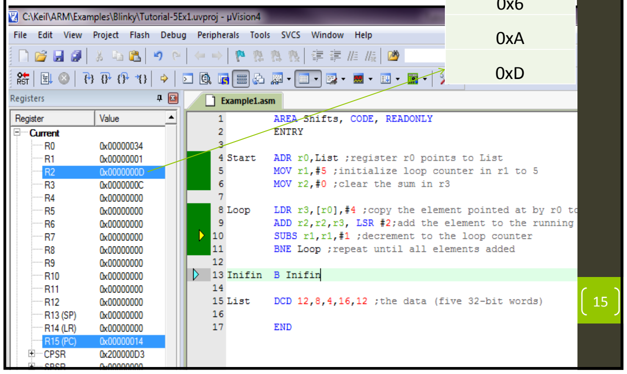

C:\Keil\ARM\Examples\Blinky\Tutorial-5Ex2.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Ex2.asm

```
Register      Value
Current
  R0          0x00000024
  R1          0x00000005
  R2          0x00000030
  R3          0x0000000C
  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)    0x00000014
  CPSR        0x000000D3
  SPSR        0x00000000
```

```
        AREA Shifts, CODE, READONLY
        ENTRY

Start   ADR r0,List ;register r0 points to List
        MOV r1,#5 ;initialize loop counter in r1 to 5
        MOV r2,#0 ;clear the sum in r3

Loop    LDR r3,[r0],#4 ;copy the element pointed at by r0 to r3 and
        ADD r2,r2,r3, LSL #2;add the element to the running total
        SUBS r1,r1,#1 ;decrement to the loop counter
        BNE Loop ;repeat until all elements added

Inifin  B Inifin

List    DCD 12,8,4,16,12 ;the data (five 32-bit words)

        END
```

Example-1: Logical Shift

0x3
0x5
0x6
0xA
0xD

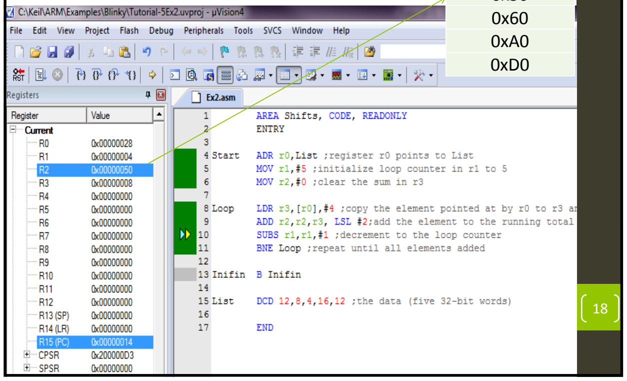

C:\Keil\ARM\Examples\Blinky\Tutorial-5Ex1.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Example1.asm

```
Register      Value
Current
  R0          0x00000034
  R1          0x00000001
  R2          0x0000000D
  R3          0x0000000C
  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)    0x00000014
  CPSR        0x200000D3
```

```
        AREA Shifts, CODE, READONLY
        ENTRY

Start   ADR r0,List ;register r0 points to List
        MOV r1,#5 ;initialize loop counter in r1 to 5
        MOV r2,#0 ;clear the sum in r3

Loop    LDR r3,[r0],#4 ;copy the element pointed at by r0 to
        ADD r2,r2,r3, LSR #2;add the element to the running
        SUBS r1,r1,#1 ;decrement to the loop counter
        BNE Loop ;repeat until all elements added

Inifin  B Inifin

List    DCD 12,8,4,16,12 ;the data (five 32-bit words)

        END
```

Example-2: Logical Shift

CumSum (hex)
0x30
0x50
0x60
0xA0
0xD0

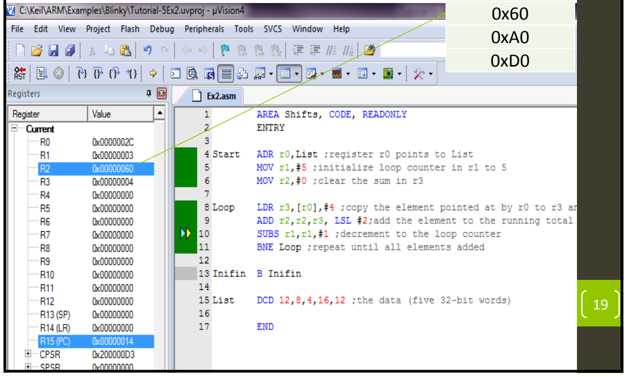

C:\Keil\ARM\Examples\Blinky\Tutorial-5Ex2.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Ex2.asm

```
Register      Value
Current
  R0          0x00000028
  R1          0x00000004
  R2          0x00000050
  R3          0x00000008
  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)    0x00000014
  CPSR        0x200000D3
  SPSR        0x00000000
```

```
        AREA Shifts, CODE, READONLY
        ENTRY

Start   ADR r0,List ;register r0 points to List
        MOV r1,#5 ;initialize loop counter in r1 to 5
        MOV r2,#0 ;clear the sum in r3

Loop    LDR r3,[r0],#4 ;copy the element pointed at by r0 to r3 ar
        ADD r2,r2,r3, LSL #2;add the element to the running total
        SUBS r1,r1,#1 ;decrement to the loop counter
        BNE Loop ;repeat until all elements added

Inifin  B Inifin

List    DCD 12,8,4,16,12 ;the data (five 32-bit words)

        END
```

Example-2: Logical Shift

CumSum (hex)
0x30
0x50
0x60
0xA0
0xD0

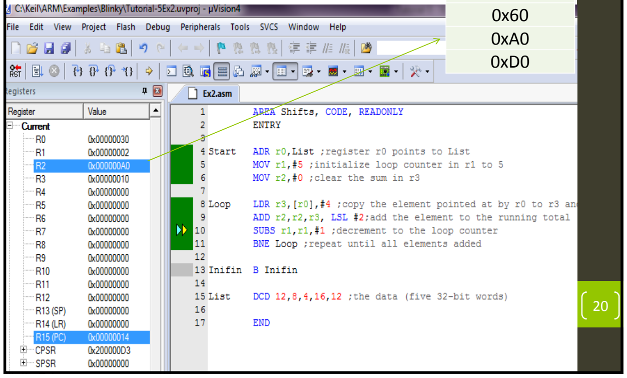

C:\Keil\ARM\Examples\Blinky\Tutorial-5Ex2.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Ex2.asm

Register	Value
Current	
R0	0x0000002C
R1	0x00000003
R2	0x00000060
R3	0x00000004
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)	0x00000014
CPSR	0x200000D3
SPSR	0x00000000

```
        AREA Shifts, CODE, READONLY
        ENTRY

Start   ADR r0,List ;register r0 points to List
        MOV r1,#5 ;initialize loop counter in r1 to 5
        MOV r2,#0 ;clear the sum in r3

Loop    LDR r3,[r0],#4 ;copy the element pointed at by r0 to r3 ar
        ADD r2,r2,r3, LSL #2;add the element to the running total
        SUBS r1,r1,#1 ;decrement to the loop counter
        BNE Loop ;repeat until all elements added

Inifin  B Inifin

List    DCD 12,8,4,16,12 ;the data (five 32-bit words)

        END
```

Example-2: Logical Shift

CumSum (hex)
0x30
0x50
0x60
0xA0
0xD0

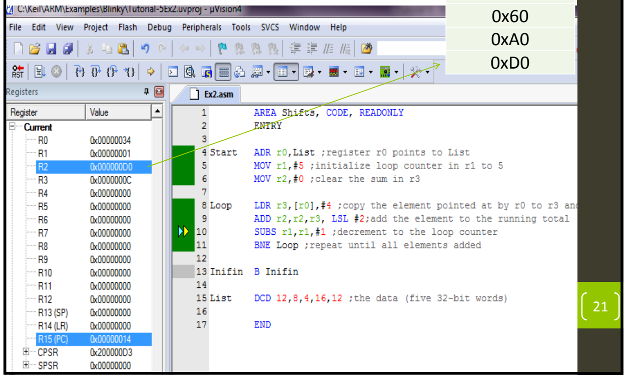

C:\Keil\ARM\Examples\Blinky\Tutorial-5Ex2.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
egisters
Ex2.asm

Register	Value
Current	
R0	0x00000030
R1	0x00000002
R2	0x000000A0
R3	0x00000010
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)	0x00000014
CPSR	0x200000D3
SPSR	0x00000000

```
        AREA Shifts, CODE, READONLY
        ENTRY

Start   ADR r0,List ;register r0 points to List
        MOV r1,#5 ;initialize loop counter in r1 to 5
        MOV r2,#0 ;clear the sum in r3

Loop    LDR r3,[r0],#4 ;copy the element pointed at by r0 to r3 and
        ADD r2,r2,r3, LSL #2;add the element to the running total
        SUBS r1,r1,#1 ;decrement to the loop counter
        BNE Loop ;repeat until all elements added

Inifin  B Inifin

List    DCD 12,8,4,16,12 ;the data (five 32-bit words)

        END
```

Example-2: Logical Shift

CumSum (hex)
0x30
0x50
0x60
0xA0
0xD0


C:\Keil\ARM\Examples\Blinky\Tutorial-5Ex2.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Ex2.asm

Register	Value
Current	
R0	0x00000034
R1	0x00000001
R2	0x000000D0
R3	0x0000000C
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)	0x00000014
CPSR	0x200000D3
SPSR	0x00000000

```
        AREA Shifts, CODE, READONLY
        ENTRY

Start   ADR r0,List ;register r0 points to List
        MOV r1,#5 ;initialize loop counter in r1 to 5
        MOV r2,#0 ;clear the sum in r3

Loop    LDR r3,[r0],#4 ;copy the element pointed at by r0 to r3 and
        ADD r2,r2,r3, LSL #2;add the element to the running total
        SUBS r1,r1,#1 ;decrement to the loop counter
        BNE Loop ;repeat until all elements added

Inifin  B Inifin

List    DCD 12,8,4,16,12 ;the data (five 32-bit words)

        END
```

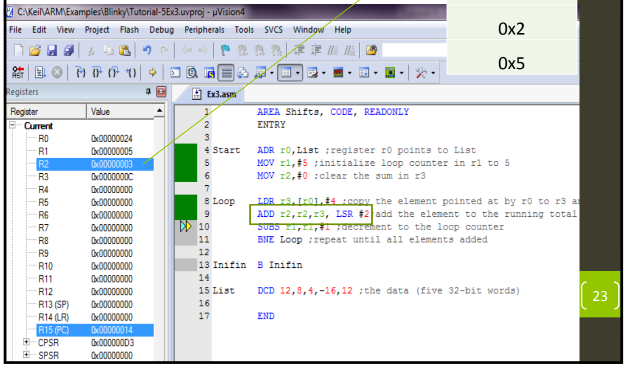
Example-3: Logical Shift

- We have a list of unsigned numbers.
- **Divide** them by 4 in an **efficient way** and sum them up.
- Example:

Number		Number/4	CumSum	CumSum (hex)
12	4	3	3	0x3
8	4	2	5	0x5
4	4	1	6	0x6
-16	4	-4	2	0x2
12	4	3	5	0x5
Sum		5		0x5

Example-3: Logical Shift

0x3
0x5
0x6
0x2
0x5

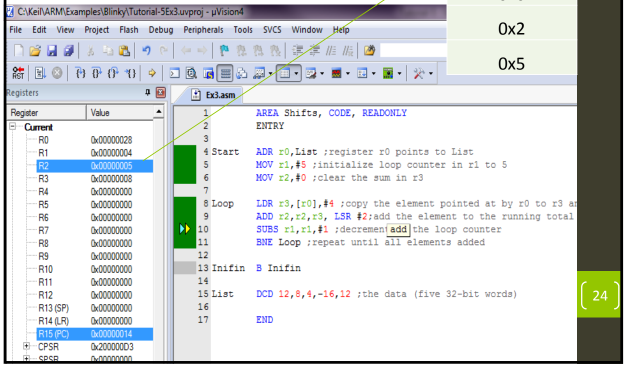

C:\Keil\ARM\Examples\Blinky\Tutorial-5Ex3.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Ex3.asm

Register	Value
Current	
R0	0x00000024
R1	0x00000005
R2	0x00000003
R3	0x0000000C
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)	0x00000014
CPSR	0x000000D3
SPSR	0x00000000

```
        AREA Shifts, CODE, READONLY
        ENTRY

Start   ADR r0,List ;register r0 points to List
        MOV r1,#5 ;initialize loop counter in r1 to 5
        MOV r2,#0 ;clear the sum in r3

Loop    LDR r3,[r0],#4 ;copy the element pointed at by r0 to r3 ar
        ADD r2,r2,r3, LSR #2;add the element to the running total
        SUBS r1,r1,#1 ;decrement to the loop counter
        BNE Loop ;repeat until all elements added

Inifin  B Inifin

List    DCD 12,8,4,-16,12 ;the data (five 32-bit words)

        END
```

Example-3: Logical Shift

0x3
0x5
0x6
0x2
0x5


C:\Keil\ARM\Examples\Blinky\Tutorial-5Ex3.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Ex3.asm
add

Register	Value
Current	
R0	0x00000028
R1	0x00000004
R2	0x00000005
R3	0x00000008
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)	0x00000014
CPSR	0x200000D3
SPSR	0x00000000

```
        AREA Shifts, CODE, READONLY
        ENTRY

Start   ADR r0,List ;register r0 points to List
        MOV r1,#5 ;initialize loop counter in r1 to 5
        MOV r2,#0 ;clear the sum in r3

Loop    LDR r3,[r0],#4 ;copy the element pointed at by r0 to r3 ar
        ADD r2,r2,r3, LSR #2;add the element to the running total
        SUBS r1,r1,#1 ;decremen the loop counter
        BNE Loop ;repeat until all elements added

Inifin  B Inifin

List    DCD 12,8,4,-16,12 ;the data (five 32-bit words)

        END
```

Example-3: Logical Shift

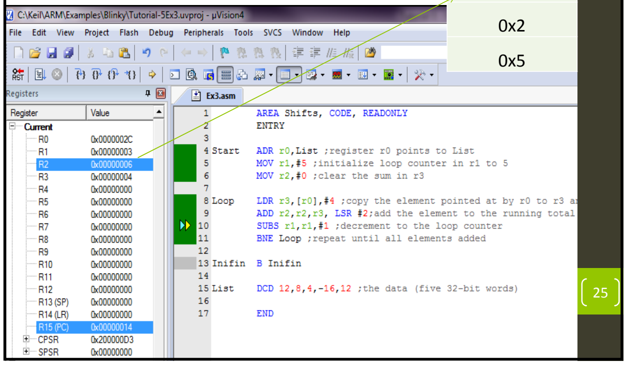
0x3

0x5

0x6

0x2

0x5


C:\Keil\ARM\Examples\Blinky\Tutorial-5Ex3.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Ex3.asm

Register	Value
Current	
R0	0x0000002C
R1	0x00000003
R2	0x00000006
R3	0x00000004
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)	0x00000014
CPSR	0x200000D3
SPSR	0x00000000

```
        AREA Shifts, CODE, READONLY
        ENTRY

Start   ADR r0,List ;register r0 points to List
        MOV r1,#5 ;initialize loop counter in r1 to 5
        MOV r2,#0 ;clear the sum in r3

Loop    LDR r3,[r0],#4 ;copy the element pointed at by r0 to r3 a
        ADD r2,r2,r3, LSR #2;add the element to the running total
        SUBS r1,r1,#1 ;decrement to the loop counter
        BNE Loop ;repeat until all elements added

Inifin  B Inifin

List    DCD 12,8,4,-16,12 ;the data (five 32-bit words)

        END
```

25

Example-3: Logical Shift

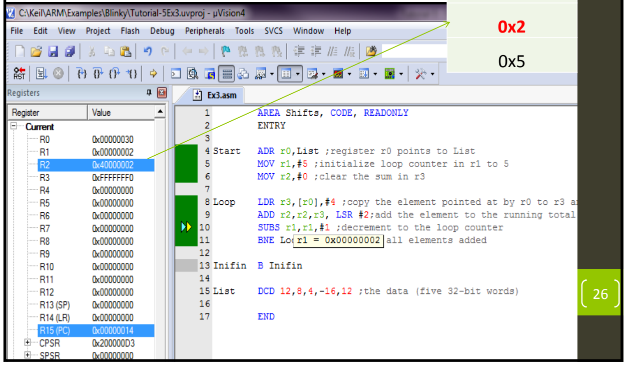
0x3

0x5

0x6

0x2

0x5


C:\Keil\ARM\Examples\Blinky\Tutorial-5Ex3.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Ex3.asm

Register	Value
Current	
R0	0x00000030
R1	0x00000002
R2	0x40000002
R3	0xFFFFFFF0
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)	0x00000014
CPSR	0x200000D3
SPSR	0x00000000

```
        AREA Shifts, CODE, READONLY
        ENTRY

Start   ADR r0,List ;register r0 points to List
        MOV r1,#5 ;initialize loop counter in r1 to 5
        MOV r2,#0 ;clear the sum in r3

Loop    LDR r3,[r0],#4 ;copy the element pointed at by r0 to r3 a
        ADD r2,r2,r3, LSR #2;add the element to the running total
        SUBS r1,r1,#1 ;decrement to the loop counter
        BNE Lo r1 = 0x00000002 all elements added

Inifin  B Inifin

List    DCD 12,8,4,-16,12 ;the data (five 32-bit words)

        END
```

26

Example-3: Logical Shift

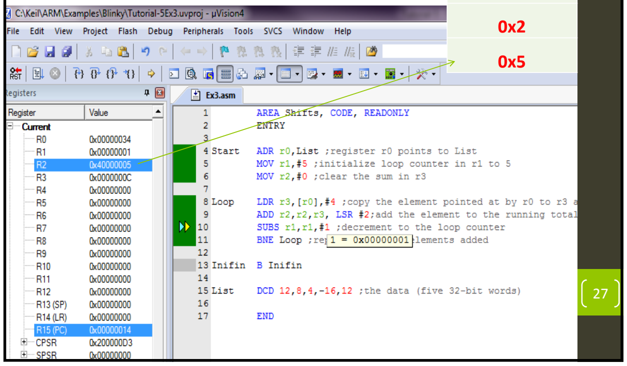
0x3

0x5

0x6

0x2

0x5


C:\Keil\ARM\Examples\Blinky\Tutorial-5Ex3.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
egisters
Ex3.asm

Register	Value
Current	
R0	0x00000034
R1	0x00000001
R2	0x40000005
R3	0x0000000C
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)	0x00000014
CPSR	0x200000D3
SPSR	0x00000000

```
        AREA Shifts, CODE, READONLY
        ENTRY

Start   ADR r0,List ;register r0 points to List
        MOV r1,#5 ;initialize loop counter in r1 to 5
        MOV r2,#0 ;clear the sum in r3

Loop    LDR r3,[r0],#4 ;copy the element pointed at by r0 to r3 a
        ADD r2,r2,r3, LSR #2;add the element to the running total
        SUBS r1,r1,#1 ;decrement to the loop counter
        BNE Loop ;rep 1 = 0x00000001 elements added

Inifin  B Inifin

List    DCD 12,8,4,-16,12 ;the data (five 32-bit words)

        END
```

27

What did you do wrong?

- Should use arithmetic shift right on signed numbers

28

Example-3: Arithmetic Shi

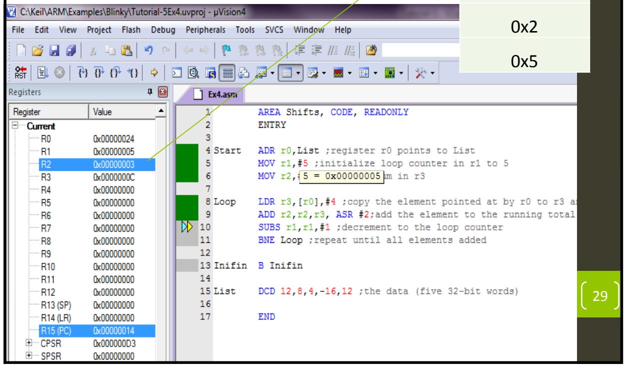
0x3

0x5

0x6

0x2

0x5


C:\Keil\ARM\Examples\Blinky\Tutorial-5Ex4.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Ex4.asm

Register	Value
Current	
R0	0x00000024
R1	0x00000005
R2	0x00000003
R3	0x0000000C
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)	0x00000014
CPSR	0x000000D3
SPSR	0x00000000

```
        AREA Shifts, CODE, READONLY
        ENTRY

Start   ADR r0,List ;register r0 points to List
        MOV r1,#5 ;initialize loop counter in r1 to 5
        MOV r2, 5 = 0x00000005 m in r3

Loop    LDR r3,[r0],#4 ;copy the element pointed at by r0 to r3 a
        ADD r2,r2,r3, ASR #2;add the element to the running total
        SUBS r1,r1,#1 ;decrement to the loop counter
        BNE Loop ;repeat until all elements added

Inifin  B Inifin

List    DCD 12,8,4,-16,12 ;the data (five 32-bit words)

        END
```

29

Example-3: Arithmetic Shi

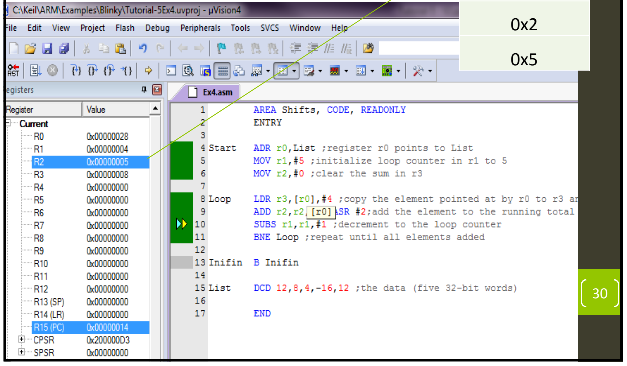
0x3

0x5

0x6

0x2

0x5


C:\Keil\ARM\Examples\Blinky\Tutorial-5Ex4.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
egisters
Ex4.asm

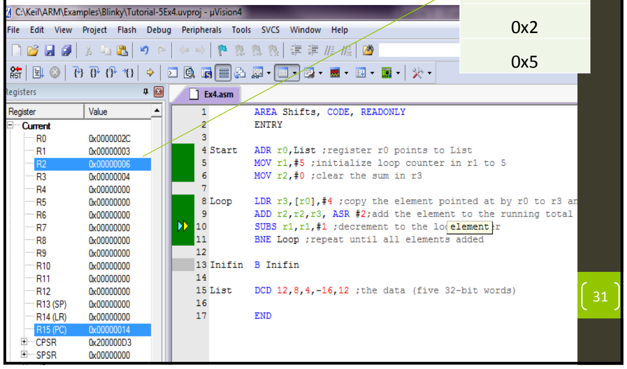
Register	Value
Current	
R0	0x00000028
R1	0x00000004
R2	0x00000005
R3	0x00000008
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x00000014
CPSR	0x200000D3
SPSR	0x00000000

```
        AREA Shifts, CODE, READONLY
        ENTRY

Start   ADR r0,List ;register r0 points to List
        MOV r1,#5 ;initialize loop counter in r1 to 5
        MOV r2,#0 ;clear the sum in r3

Loop    LDR r3,[r0],#4 ;copy the element pointed at by r0 to r3 a
        ADD r2,r2, [r0] ASR #2;add the element to the running total
        SUBS r1,r1,#1 ;decrement to the loop counter
        BNE Loop ;repeat until all elements added

Inifin  B Inifin

List    DCD 12,8,4,-16,12 ;the data (five 32-bit words)

        END
```

30

Example-3: Arithmetic Shi

0x3
0x5
0x6
0x2
0x5

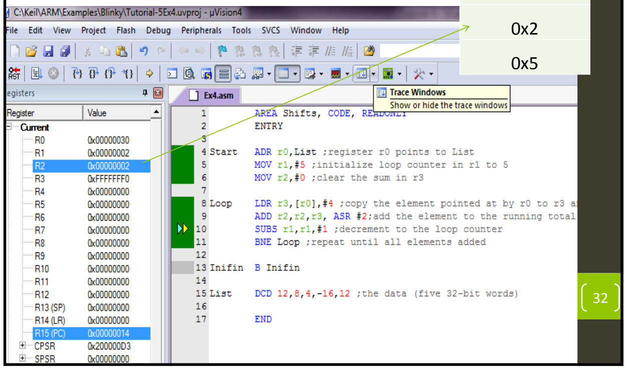

C:\Keil\ARM\Examples\Blinky\Tutorial-5Ex4.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Ex4.asm

Register	Value
Current	
R0	0x0000002C
R1	0x00000003
R2	0x00000006
R3	0x00000004
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)	0x00000014
CPSR	0x200000D3
SPSR	0x00000000

```
        AREA Shifts, CODE, READONLY
        ENTRY

Start   ADR r0,List ;register r0 points to List
        MOV r1,#5 ;initialize loop counter in r1 to 5
        MOV r2,#0 ;clear the sum in r3

Loop    LDR r3,[r0],#4 ;copy the element pointed at by r0 to r3 an
        ADD r2,r2,r3, ASR #2;add the element to the running total
        SUBS r1,r1,#1 ;decrement to the lo element r
        BNE Loop ;repeat until all elements added

Inifin  B Inifin

List    DCD 12,8,4,-16,12 ;the data (five 32-bit words)

        END
```

Example-3: Arithmetic Shi

0x3
0x5
0x6
0x2
0x5

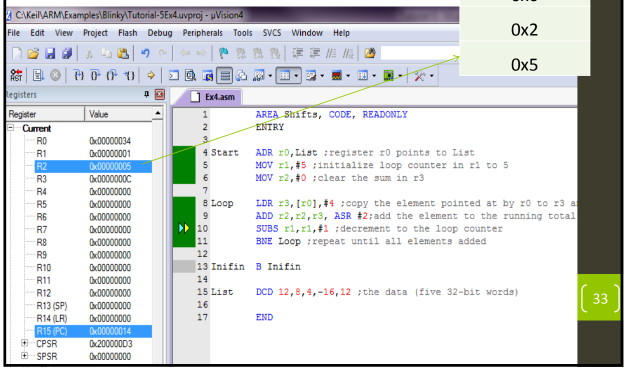

C:\Keil\ARM\Examples\Blinky\Tutorial-5Ex4.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Ex4.asm
Trace Windows
Show or hide the trace windows

Register	Value
Current	
R0	0x00000030
R1	0x00000002
R2	0x00000002
R3	0xFFFFFFF0
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)	0x00000014
CPSR	0x200000D3
SPSR	0x00000000

```
        AREA Shifts, CODE, READONLY
        ENTRY

Start   ADR r0,List ;register r0 points to List
        MOV r1,#5 ;initialize loop counter in r1 to 5
        MOV r2,#0 ;clear the sum in r3

Loop    LDR r3,[r0],#4 ;copy the element pointed at by r0 to r3 a
        ADD r2,r2,r3, ASR #2;add the element to the running total
        SUBS r1,r1,#1 ;decrement to the loop counter
        BNE Loop ;repeat until all elements added

Inifin  B Inifin

List    DCD 12,8,4,-16,12 ;the data (five 32-bit words)

        END
```

Example-3: Arithmetic Shi

0x3
0x5
0x6
0x2
0x5


C:\Keil\ARM\Examples\Blinky\Tutorial-5Ex4.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Ex4.asm

Register	Value
Current	
R0	0x00000034
R1	0x00000001
R2	0x00000005
R3	0x0000000C
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)	0x00000014
CPSR	0x200000D3
SPSR	0x00000000

```
        AREA Shifts, CODE, READONLY
        ENTRY

Start   ADR r0,List ;register r0 points to List
        MOV r1,#5 ;initialize loop counter in r1 to 5
        MOV r2,#0 ;clear the sum in r3

Loop    LDR r3,[r0],#4 ;copy the element pointed at by r0 to r3 a
        ADD r2,r2,r3, ASR #2;add the element to the running total
        SUBS r1,r1,#1 ;decrement to the loop counter
        BNE Loop ;repeat until all elements added

Inifin  B Inifin

List    DCD 12,8,4,-16,12 ;the data (five 32-bit words)

        END
```

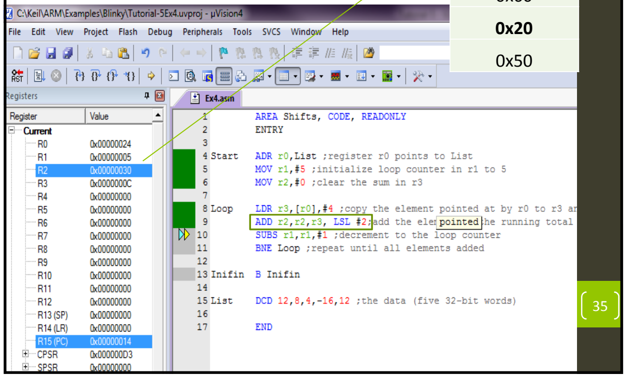
Example-4: Logical Shift

- We have a list of numbers.
- **Multiply** them by 4 in an **efficient way** and sum them up.
- Example:

Number		NumberX4	CumSum	CumSum (hex)
12	4	48	48	0x30
8	4	32	80	0x50
4	4	16	96	0x60
-16	**4**	**-64**	**32**	**0x20**
12	4	48	80	0x50
Sum		**80**		**0x50**

Example-4: Logical Shift

0x30
0x50
0x60
0x20
0x50

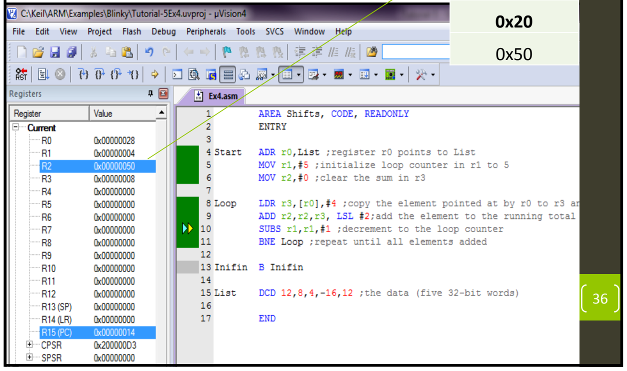

C:\Keil\ARM\Examples\Blinky\Tutorial-5Ex4.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Ex4.asm

Register	Value
Current	
R0	0x00000024
R1	0x00000005
R2	0x00000030
R3	0x0000000C
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)	0x00000014
CPSR	0x000000D3
SPSR	0x00000000

```
        AREA Shifts, CODE, READONLY
        ENTRY

Start   ADR r0,List ;register r0 points to List
        MOV r1,#5 ;initialize loop counter in r1 to 5
        MOV r2,#0 ;clear the sum in r3

Loop    LDR r3,[r0],#4 ;copy the element pointed at by r0 to r3 an
        ADD r2,r2,r3, LSL #2;add the ele pointed the running total
        SUBS r1,r1,#1 ;decrement to the loop counter
        BNE Loop ;repeat until all elements added

Inifin  B Inifin

List    DCD 12,8,4,-16,12 ;the data (five 32-bit words)

        END
```

Example-4: Logical Shift

0x30
0x50
0x60
0x20
0x50


C:\Keil\ARM\Examples\Blinky\Tutorial-5Ex4.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Ex4.asm

Register	Value
Current	
R0	0x00000028
R1	0x00000004
R2	0x00000050
R3	0x00000008
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)	0x00000014
CPSR	0x200000D3
SPSR	0x00000000

```
        AREA Shifts, CODE, READONLY
        ENTRY

Start   ADR r0,List ;register r0 points to List
        MOV r1,#5 ;initialize loop counter in r1 to 5
        MOV r2,#0 ;clear the sum in r3

Loop    LDR r3,[r0],#4 ;copy the element pointed at by r0 to r3 an
        ADD r2,r2,r3, LSL #2;add the element to the running total
        SUBS r1,r1,#1 ;decrement to the loop counter
        BNE Loop ;repeat until all elements added

Inifin  B Inifin

List    DCD 12,8,4,-16,12 ;the data (five 32-bit words)

        END
```

Example-4: Logical Shift

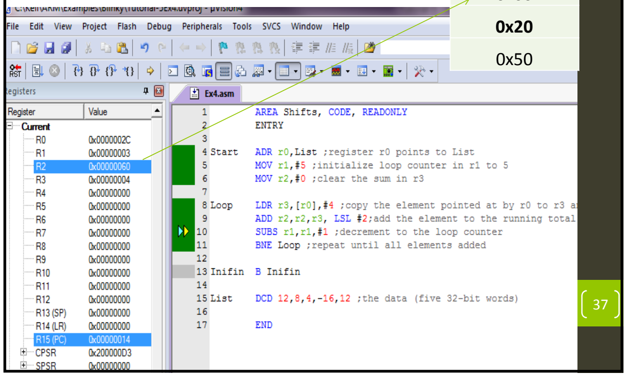
0x30

0x50

0x60

0x20

0x50



C:\Keil\ARM\Examples\Blinky\Tutorial-5Ex4.uvproj - µVision4

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Ex4.asm

Register	Value
Current	
R0	0x0000002C
R1	0x00000003
R2	0x00000060
R3	0x00000004
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)	0x00000014
CPSR	0x200000D3
SPSR	0x00000000

```
        AREA Shifts, CODE, READONLY
        ENTRY

Start   ADR r0,List ;register r0 points to List
        MOV r1,#5 ;initialize loop counter in r1 to 5
        MOV r2,#0 ;clear the sum in r3

Loop    LDR r3,[r0],#4 ;copy the element pointed at by r0 to r3 a
        ADD r2,r2,r3, LSL #2;add the element to the running total
        SUBS r1,r1,#1 ;decrement to the loop counter
        BNE Loop ;repeat until all elements added

Inifin  B Inifin

List    DCD 12,8,4,-16,12 ;the data (five 32-bit words)

        END
```

No ASL in ARM

- There's no ASL (Arithmetic Shift Left) in ARM.
- It's safe to use LSL instead (both are identical)

Example-4: Logical Shift

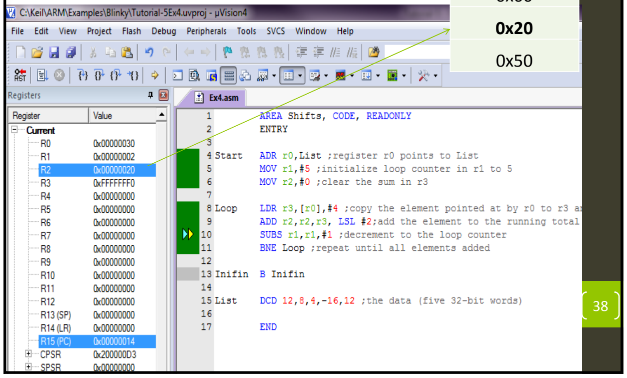
0x30

0x50

0x60

0x20

0x50



C:\Keil\ARM\Examples\Blinky\Tutorial-5Ex4.uvproj - µVision4

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Ex4.asm

Register	Value
Current	
R0	0x00000030
R1	0x00000002
R2	0x00000020
R3	0xFFFFFFF0
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)	0x00000014
CPSR	0x200000D3
SPSR	0x00000000

```
        AREA Shifts, CODE, READONLY
        ENTRY

Start   ADR r0,List ;register r0 points to List
        MOV r1,#5 ;initialize loop counter in r1 to 5
        MOV r2,#0 ;clear the sum in r3

Loop    LDR r3,[r0],#4 ;copy the element pointed at by r0 to r3 a
        ADD r2,r2,r3, LSL #2;add the element to the running total
        SUBS r1,r1,#1 ;decrement to the loop counter
        BNE Loop ;repeat until all elements added

Inifin  B Inifin

List    DCD 12,8,4,-16,12 ;the data (five 32-bit words)

        END
```

What we learned so far

- To multiply numbers, we should use:
  - LSL: Logical Shift Left
  - There's no arithmetic shift left
- To divide numbers, we should use:
  - ASR: Arithmetic Shift right
  - **Logical Shift Left will give wrong results**

Example-4: Logical Shift

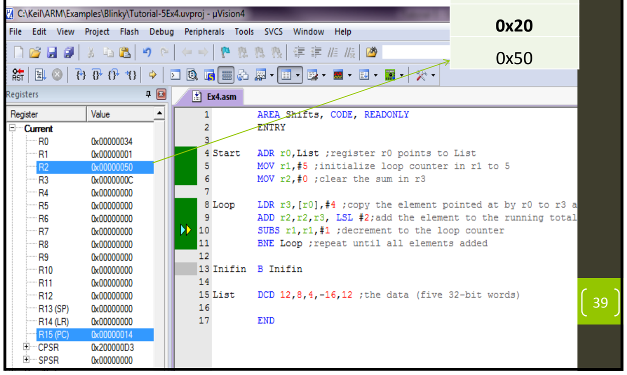
0x30

0x50

0x60

0x20

0x50



C:\Keil\ARM\Examples\Blinky\Tutorial-5Ex4.uvproj - µVision4

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Ex4.asm

Register	Value
Current	
R0	0x00000034
R1	0x00000001
R2	0x00000050
R3	0x0000000C
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)	0x00000014
CPSR	0x200000D3
SPSR	0x00000000

```
        AREA Shifts, CODE, READONLY
        ENTRY

Start   ADR r0,List ;register r0 points to List
        MOV r1,#5 ;initialize loop counter in r1 to 5
        MOV r2,#0 ;clear the sum in r3

Loop    LDR r3,[r0],#4 ;copy the element pointed at by r0 to r3 a
        ADD r2,r2,r3, LSL #2;add the element to the running total
        SUBS r1,r1,#1 ;decrement to the loop counter
        BNE Loop ;repeat until all elements added

Inifin  B Inifin

List    DCD 12,8,4,-16,12 ;the data (five 32-bit words)

        END
```

Example-6: Rotate Shift

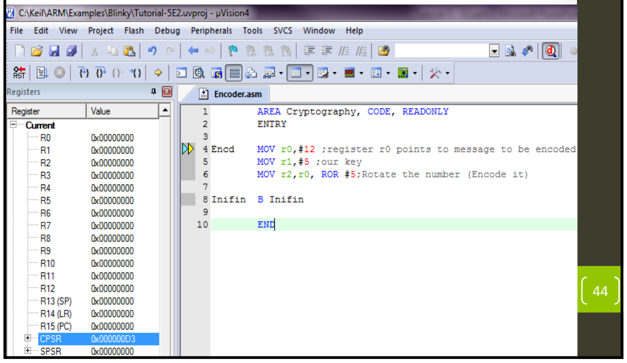
- Assume a person wants to send a secure number to another person.
- They agree to use rotate shift to encrypt the message

secure
message
delivery

Example-6: Rotate Shift

- Assume the message sent is the number: 12
- Assume the mutually agreed key is: 5

Example-6: Rotate Shift-Encoder

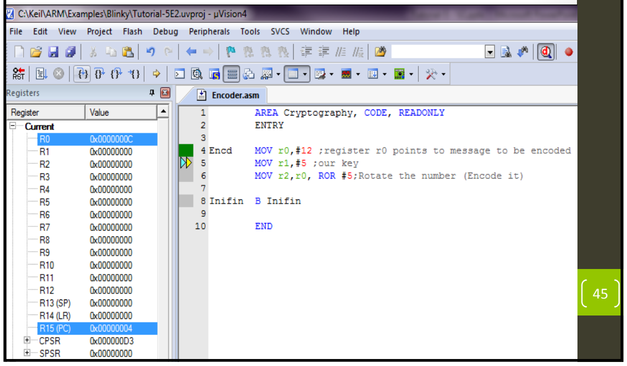


C:\Keil\ARM\Examples\Blinky\Tutorial-5E2.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Encoder.asm

Register	Value
Current	
R0	0x00000000
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)	0x00000000
CPSR	0x000000D3
SPSR	0x00000000

```
        AREA Cryptography, CODE, READONLY
        ENTRY

Encd    MOV r0,#12 ;register r0 points to message to be encoded
        MOV r1,#5 ;our key
        MOV r2,r0, ROR #5;Rotate the number (Encode it)

Inifin  B Inifin

        END
```

Example-6: Rotate Shift-Encoder

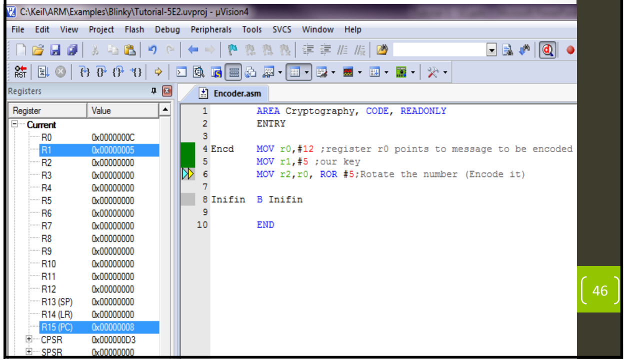


C:\Keil\ARM\Examples\Blinky\Tutorial-5E2.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Encoder.asm

Register	Value
Current	
R0	0x0000000C
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

```
        AREA Cryptography, CODE, READONLY
        ENTRY

Encd    MOV r0,#12 ;register r0 points to message to be encoded
        MOV r1,#5 ;our key
        MOV r2,r0, ROR #5;Rotate the number (Encode it)

Inifin  B Inifin

        END
```

Example-6: Rotate Shift-Encoder

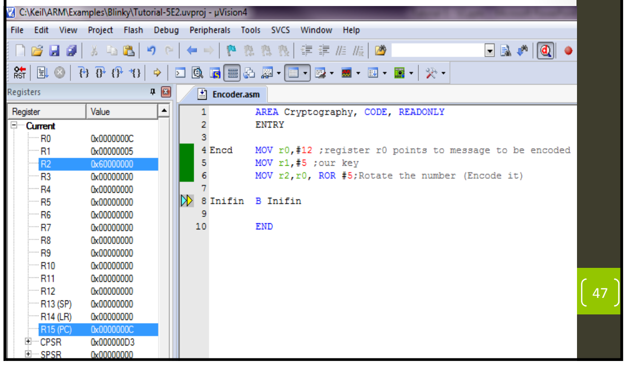


C:\Keil\ARM\Examples\Blinky\Tutorial-5E2.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Encoder.asm

Register	Value
Current	
R0	0x0000000C
R1	0x00000005
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

```
        AREA Cryptography, CODE, READONLY
        ENTRY

Encd    MOV r0,#12 ;register r0 points to message to be encoded
        MOV r1,#5 ;our key
        MOV r2,r0, ROR #5;Rotate the number (Encode it)

Inifin  B Inifin

        END
```

Example-6: Rotate Shift-Encoder



C:\Keil\ARM\Examples\Blinky\Tutorial-5E2.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Encoder.asm

Register	Value
Current	
R0	0x0000000C
R1	0x00000005
R2	0x60000000
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

```
        AREA Cryptography, CODE, READONLY
        ENTRY

Encd    MOV r0,#12 ;register r0 points to message to be encoded
        MOV r1,#5 ;our key
        MOV r2,r0, ROR #5;Rotate the number (Encode it)

Inifin  B Inifin

        END
```

Example-5: Rotate Shift

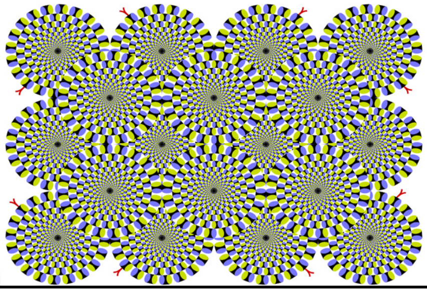
- After Encoding
  - 12 → 0x60000000 $(1610612736)_{10}$

Example-5: Rotate Shift

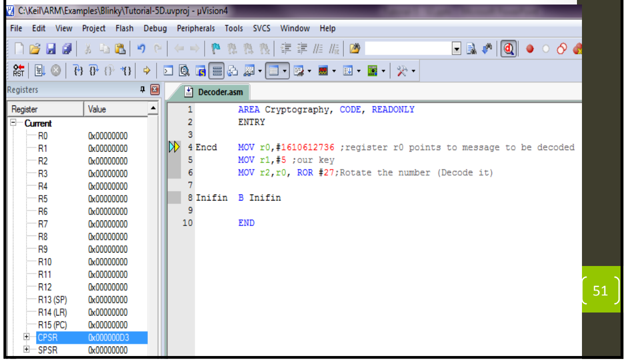
- Now, we will design the decoder to decrypt the message
- We used ROR.
  - So, we should use ROL.
- Problem. No ROL, in ARM
- How can we solve this?

Example-5: Rotate Shift

- 1 ROL = 32-1 = 31 ROR
- 5 ROL = 32-5 = 27 ROR



Example-5: Rotate Shift Decoder

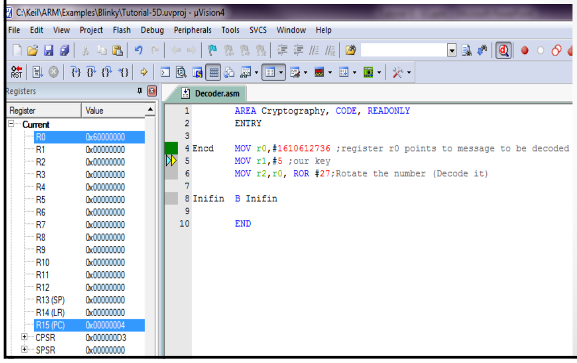


C:\Keil\ARM\Examples\Blinky\Tutorial-5D.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Decoder.asm

Register	Value
Current	
R0	0x00000000
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)	0x00000000
CPSR	0x000000D3
SPSR	0x00000000

```
        AREA Cryptography, CODE, READONLY
        ENTRY

Encd    MOV r0,#1610612736 ;register r0 points to message to be decoded
        MOV r1,#5 ;our key
        MOV r2,r0, ROR #27;Rotate the number (Decode it)

Inifin  B Inifin

        END
```

Example-5: Rotate Shift Decoder

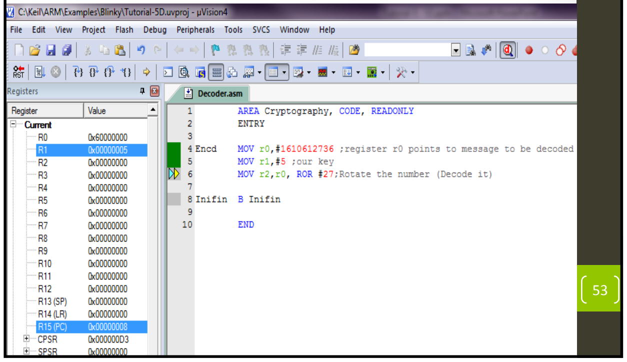


C:\Keil\ARM\Examples\Blinky\Tutorial-5D.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Decoder.asm

Register	Value
Current	
R0	0x60000000
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

```
        AREA Cryptography, CODE, READONLY
        ENTRY

Encd    MOV r0,#1610612736 ;register r0 points to message to be decoded
        MOV r1,#5 ;our key
        MOV r2,r0, ROR #27;Rotate the number (Decode it)

Inifin  B Inifin

        END
```

Example-5: Rotate Shift Decoder

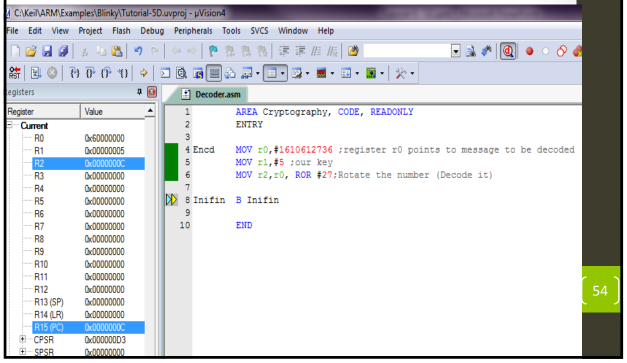


C:\Keil\ARM\Examples\Blinky\Tutorial-5D.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Decoder.asm

Register	Value
Current	
R0	0x60000000
R1	0x00000005
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

```
        AREA Cryptography, CODE, READONLY
        ENTRY

Encd    MOV r0,#1610612736 ;register r0 points to message to be decoded
        MOV r1,#5 ;our key
        MOV r2,r0, ROR #27;Rotate the number (Decode it)

Inifin  B Inifin

        END
```

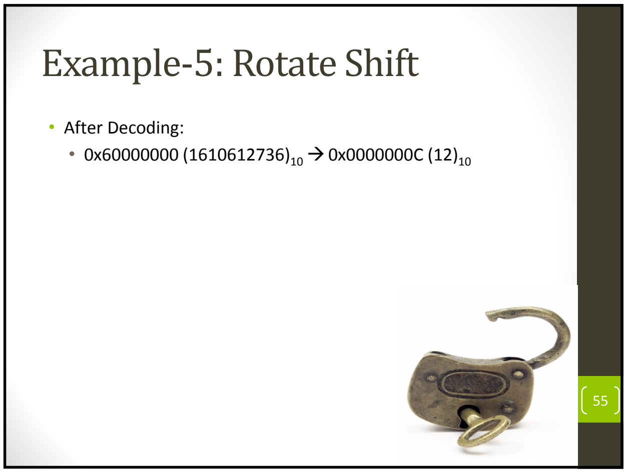
Example-5: Rotate Shift Decoder



C:\Keil\ARM\Examples\Blinky\Tutorial-5D.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
egisters
Decoder.asm

Register	Value
Current	
R0	0x60000000
R1	0x00000005
R2	0x0000000C
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

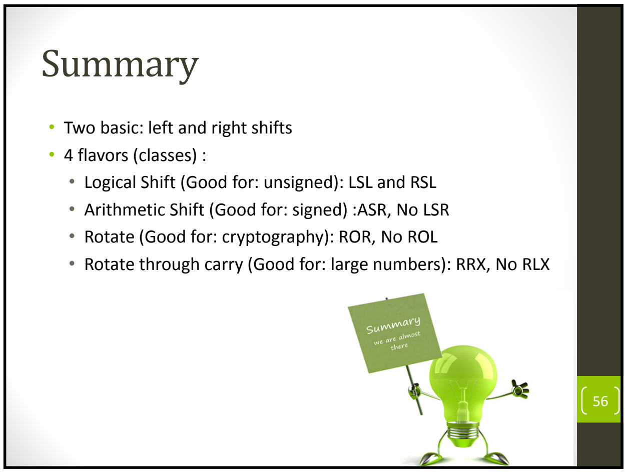
```
        AREA Cryptography, CODE, READONLY
        ENTRY

Encd    MOV r0,#1610612736 ;register r0 points to message to be decoded
        MOV r1,#5 ;our key
        MOV r2,r0, ROR #27;Rotate the number (Decode it)

Inifin  B Inifin

        END
```



Example-5: Rotate Shift

- After Decoding:
  - 0x60000000 $(1610612736)_{10}$ → 0x0000000C $(12)_{10}$

55



Summary

- Two basic: left and right shifts
- 4 flavors (classes) :
  - Logical Shift (Good for: unsigned): LSL and RSL
  - Arithmetic Shift (Good for: signed) :ASR, No LSR
  - Rotate (Good for: cryptography): ROR, No ROL
  - Rotate through carry (Good for: large numbers): RRX, No RLX

Summary
we are almost there

56