

Logical Operations

In some applications it is necessary to manipulate other sizes of data, or perhaps only individual bits.

There are instructions that perform logical AND, OR, XOR and NOT operations. The condition code register changes according to the result

AND.w d1,d2 (logically ANDs individual bits of d1 and d2 regs)

ANDi.w #\$5FFF, sr (resets trace and supervisor bits)

ANDi.w #\$FFFC, d1 (resets two low order bits of d1)

The operations can be extended to byte and longword in a similar way

Logical Operations cont.

OR.w d1,d2 (logically ORs individual bits of d1 and d2 regs)

ORi.l #\$F000000F, d0 (sets first four and last four bits of d0)

ORi.l #\$80000001, d0 (sets bits 31 and 0 while retaining others)

EORi.b #\$35, d1 (selected bits are complemented)

NOT.w d1 (each bit is complemented)

Shift and Rotate Operations

Shift instructions – left or right shift (for bit lengths 8, 16, or 32)

- Arithmetic Shifts (asl, asr)
 - If sign of data before shift and after shift are different, overflow bit is set otherwise reset.
- Logical Shifts (lsl, lsr)
 - Overflow bit is always cleared

Rotate instructions – left or right (bits not lost like in shift operations)

- Simple Rotation (rol, ror)
- Extended Rotation (roxl, roxr)

Example: left logical shift by a count of 1 of the 64-bit string in d1 and d2

```
lsl.l #1, d2
```

```
roxl.l #1, d1
```

TABLE 7.53 asl immediate instructions

Arithmetic left shifts					
Group 1: Data register shifts					
Explicit count					
Action					
Arithmetically left shift contents of indicated data register by count ranging between 1 and 8, as indicated by the IData ₃ field.					
Mnemonic	Opcode	User byte (X,N,Z,V,C bits)	Operands	Format	Length
asl.b	s74 ₇	M M M M M	IData ₃ ,dn	F18	1
asl.w	s74 ₇	M M M M M	IData ₃ ,dn	F18	1
asl.l	s74 ₇	M M M M M	Data ₃ ,dn	F18	1
Implicit count					
Action					
Arithmetically left shift contents of the indicated data register by count ranging between 0 and 63, as indicated by the contents (modulo 64) of data register Dm field.					
Mnemonic	Opcode	User byte (X,N,Z,V,C bits)	Operands	Format	Length
asl.b	s74 ₇	M M M M M	dm, dn	F18	1
asl.w	s74 ₇	M M M M M	dm, dn	F18	1
asl.l	s74 ₇	M M M M M	dm, dn	F18	1
Group 2: Memory (word) shifts					
Action					
Arithmetically left shift word indicated by memory address dmem by 1 bit.					
Mnemonic	Opcode	User byte (X,N,Z,V,C bits)	Operands	Format	Length
asl	s387 ₁₀	M M M M M	dmem	F16	1, 2, 3

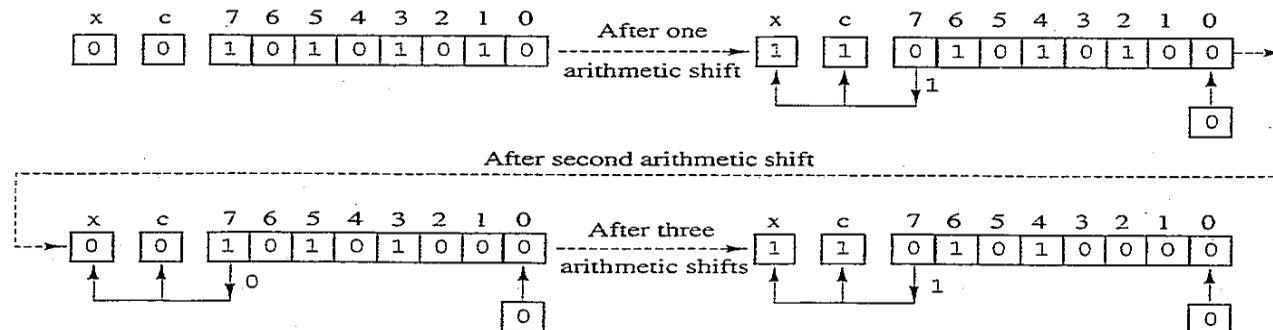


Figure 7.58 Effect of arithmetic left shift with a count of three.

asl.b #3, d2

TABLE 7.54 asr instructions

Arithmetic right shifts					
Group 1: Data register shifts					
Explicit count					
Action					
Arithmetically right shift contents of indicated data register by count ranging between 1 and 8, as indicated by $IData_2$ field.					
Mnemonic	Opcode	User byte (X.N.Z.V.C bits)	Operands	Format	Length
asr.b	$s70_7$	$\boxed{MIMIMIMIM}$	$IData_2, dn$	FIS	1
asr.w	$s70_7$	$\boxed{MIMIMIMIM}$	$IData_2, dn$	FIS	1
asr.l	$s70_7$	$\boxed{MIMIMIMIM}$	$IData_2, dn$	FIS	1
Implicit count					
Action					
Arithmetically right shift contents of indicated data register by count ranging between 0 and 63, as indicated by contents (modulo 64) of data register Dm field.					
Mnemonic	Opcode	User byte (X.N.Z.V.C bits)	Operands	Format	Length
asr.b	$s70_7$	$\boxed{MIMIMIMIM}$	dm, dn	FIS	1
asr.w	$s70_7$	$\boxed{MIMIMIMIM}$	dm, dn	FIS	1
asr.l	$s70_7$	$\boxed{MIMIMIMIM}$	dm, dn	FIS	1
Group 2: Memory (word) shifts					
Action					
Arithmetically right shift word indicated by memory address $dmem$ by 1 bit.					
Mnemonic	Opcode	User byte (X.N.Z.V.C bits)	Operands	Format	Length
asr	$s383_{10}$	$\boxed{MIMIMIMIM}$	$dmem$	F16	1, 2, 3

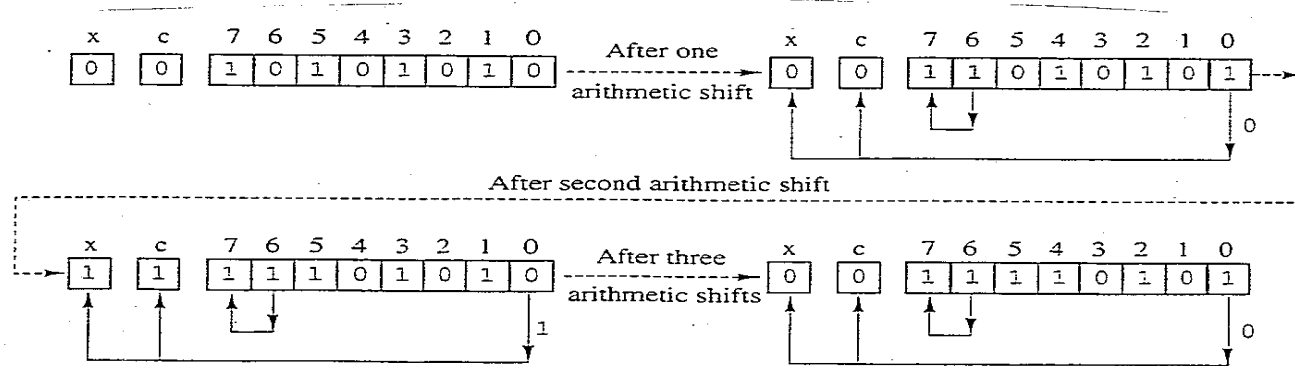


Figure 7.59 Effect of arithmetic right shift with count of three.

logical shift left lsl.e

lsl.b #3,d1

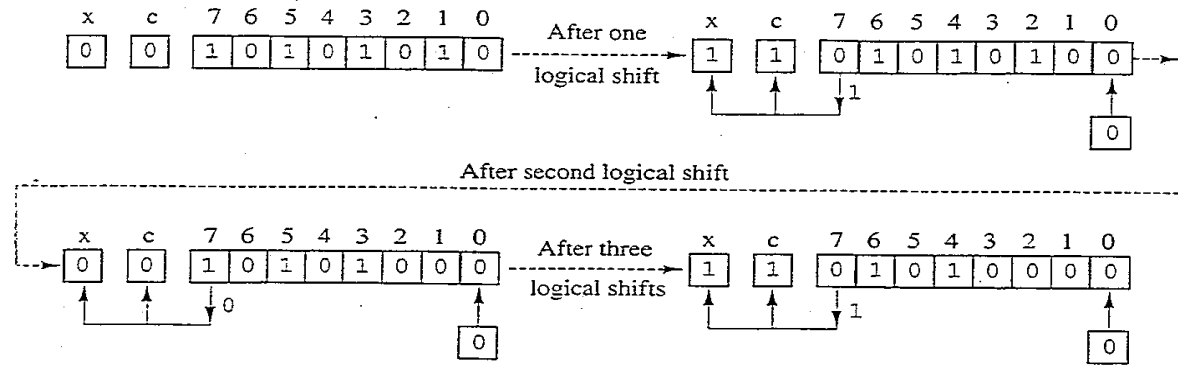


Figure 7.60 Effect of logical left shift with count of three.

logical shift right lsr.e

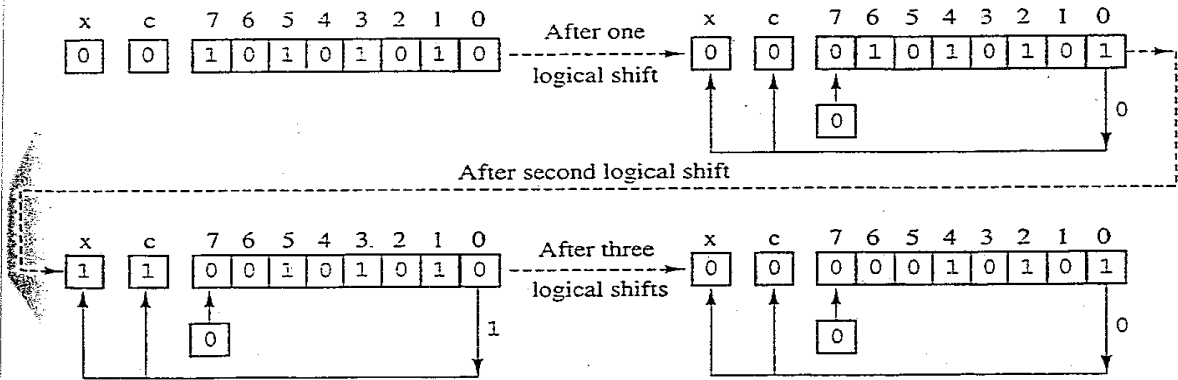


Figure 7.61 Effect of logical right shift with count of three.

lsr.b #3,d1

Left rotation ror.e

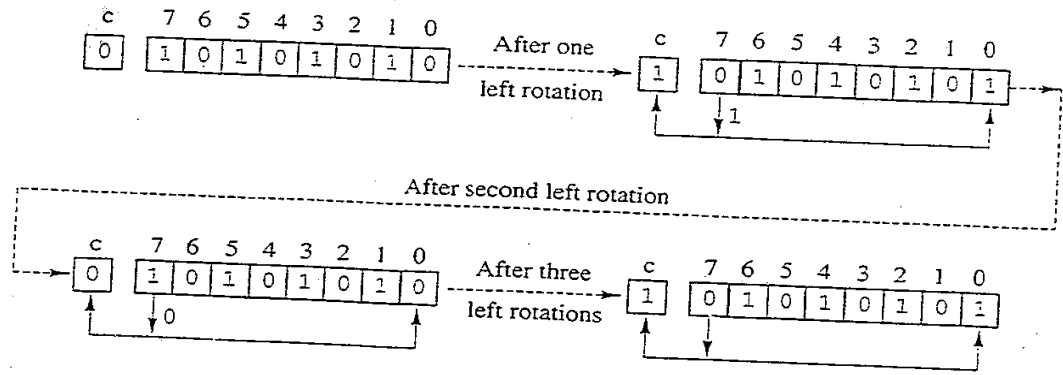


Figure 7.62 Effect of left rotation with count of three.

Right rotation ror.e

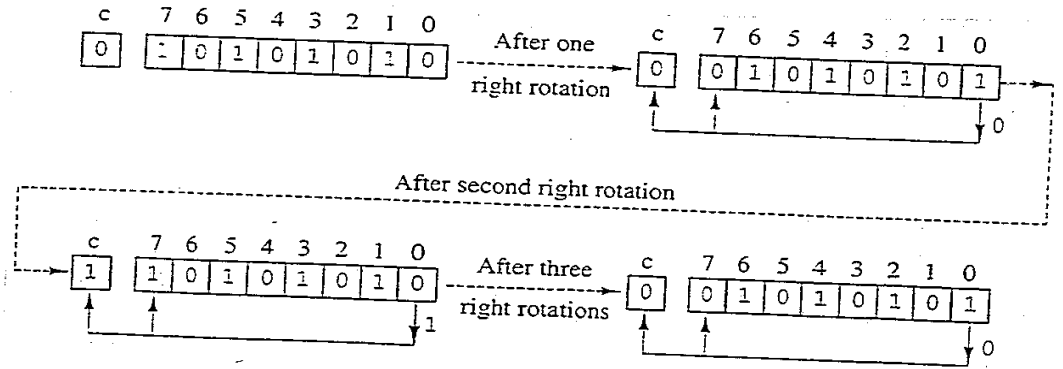
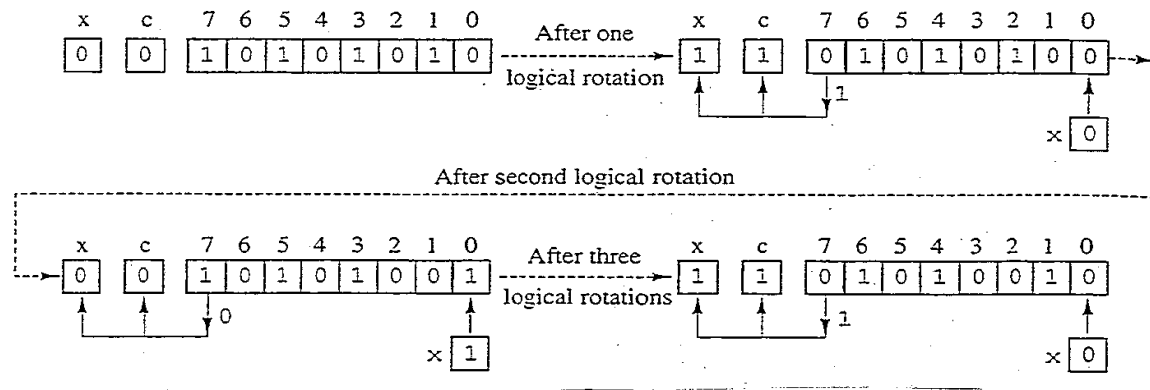


Figure 7.63 Effect of right rotation with count of three.

Extended Logical Left rotation $n \times l.e$



Extended Logical Right rotation $n \times r.e$

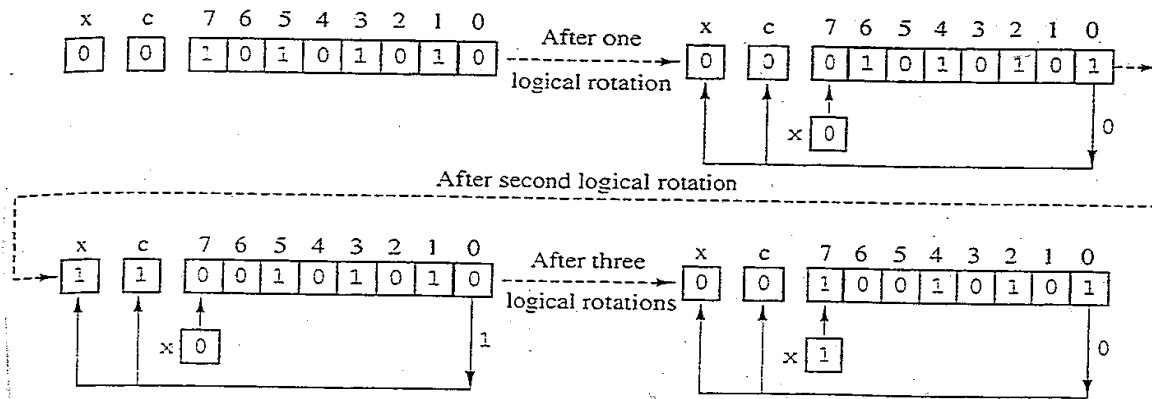


Figure 7.65 Effect of extended logical right rotation with count of three. Note that bit marked by * contains previous value of extend bit.