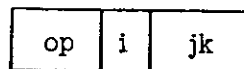


SHIFT OPERATIONS

There are three functionally different shift operations: logical shift, insert field, and integer shift. Within the logical and integer shift classes either single or double length operands may be used. There also are two ways of specifying the direction and amount of shift: either directly from a literal field in the instruction or indirectly by the contents of a register.

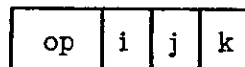
Shift Amount and Direction

When the shift amount is specified by the literal field of the instruction, the following instruction format is used:



The 10-bit literal jk-field is interpreted as a 2's complement integer, so that numbers in the range -512 to +511 are representable.

When the shift amount is specified by the contents of a register, the following instruction format is used:



The contents of register A^k or X^k is interpreted as a 2's complement integer. Only the low order 10 bits are used to specify the shift amount; the remaining bits are ignored.

The integer specifies both the direction and amount of the shift. Its absolute value specifies the amount of the shift. Its sign indicates the shift direction: a positive integer specifies a left shift, a negative integer specifies a right shift.

For the insert field instructions register A^k or X^k contains three 8-bit parameters.

Source and Result Operands

When the shift is specified by the literal field, the i-field specifies both the source and result operands; that is,

Source operand: R^i or $R^{i,i+1}$

Result operand: R^i or $R^{i,i+1}$

(where R may be interpreted as either X or A).

When the shift amount is specified by R^k , the i- and j-fields specify the source and result operands as follows:

Source operand: R^j or $R^{j,j+1}$

Result operand: R^i or $R^{i,i+1}$

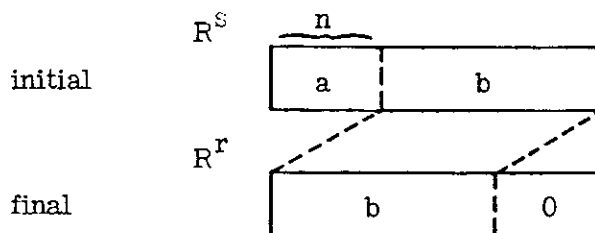
In the explanations, R^S is used to indicate the source register, and R^R the result register. The shift amount is denoted by n. The notation 48/24 is to be interpreted as 48 for the A-unit shift instructions and 24 for the X-unit shift instructions.

Logical Shift, Single Register

The contents of register R^S are shifted left or right the specified number of bit positions. The direction of the shift is determined by the sign of the shift amount. Bits which are shifted out of R^S are lost; vacated positions are filled with 0's. The 48/24-bit shifted quantity then replaces the contents of register R^R . The contents of register R^S are unchanged unless due to the operation type or the specification of the i and j field, R^S is the same register as R^R . If the shift amount is greater than or equal to 48/24, register R^R is set to 0's.

Pictorially the logical shift, single register, instructions are:

single shift left



ADVANCED COMPUTING SYSTEMS

Volume : 1A
Chapter : 02
Section : Appendix

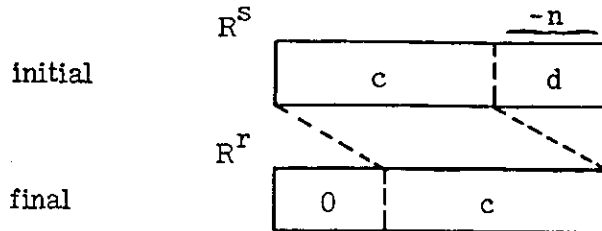
IBM REGISTERED CONFIDENTIAL

ACS-I Development Workbook

Page: 7-3

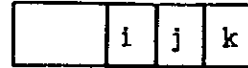
Date: 4/17/67

single shift right



Logic Shift, Arithmetic

SHA



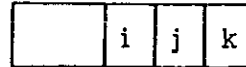
shift amount ← $A_{38, \dots, 47}^k$

A^i ← logic shift (A^j)

Exceptions: none

Logic Shift, Index

SHX



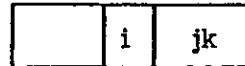
shift amount ← $X_{14, \dots, 23}^k$

X^i ← logic shift (X^j)

Exceptions: none

Logic Shift, by Constant, Arithmetic

SHAC



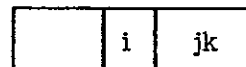
shift amount ← jk

A^i ← logic shift (A^i)

Exceptions: none

Logic Shift, by Constant, Index

SHXC



shift amount ← jk

X^i ← logic shift (X^i)

Exceptions: none

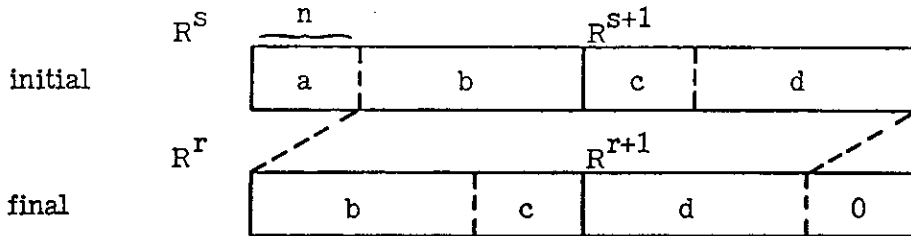
Logical Shift, Double Registers

Registers R^s and R^{s+1} are coupled and are considered as one 96/48 bit quantity. This 96/48 bit quantity is shifted left or right the specified number of bit positions to form an intermediate result. The direction of the shift is determined by the sign of the shift quantity. Bits which are shifted out are ignored; vacated positions are filled with 0's. The 96-bit intermediate result then replaces the contents of registers R^r and R^{r+1} .

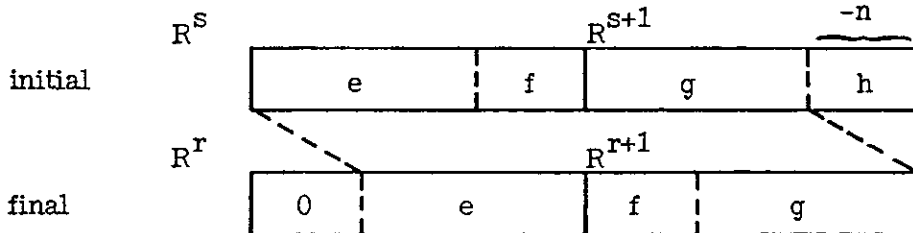
If the shift amount is greater than or equal to 96/48, registers R^r and R^{r+1} are set to 0's.

Pictorially, the logical shift, double registers, instructions are as follows:

double shift left



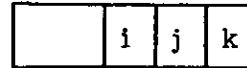
double shift right



The value of s must be even. If it is not, the low order bit specifying s is forced to 0, exception bit RS is set, and the operation proceeds.

Logic Shift, Double Arithmetic

SHD



shift amount + $A_{38, \dots, 47}^k$

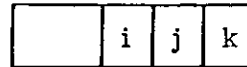
$A^{i, i+1}$ + logic shift ($A^{j, j+1}$)

Exception
 i or j odd

Exception bit
 RS

Logic Shift, Double Index

SHDX



shift amount + $X_{14, \dots, 23}^k$

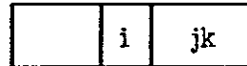
$X^{i, i+1}$ + logic shift ($X^{j, j+1}$)

Exception
 i or j odd

Exception bit
 RS

Logic Shift by Constant, Double Arithmetic

SHDC



shift amount + jk

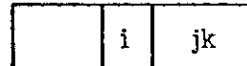
$A^{i, i+1}$ + logic shift ($A^{i, i+1}$)

Exception
 i odd

Exception bit
 RS

Logic Shift by Constant, Double Index

SHDXC



shift amount + jk

$X^{i, i+1}$ + logic shift ($X^{i, i+1}$)

Exception
 i odd

Exception bit
 RS

ADVANCED COMPUTING SYSTEMS

Volume : 1A
Chapter : 02
Section : Appendix

IBM REGISTERED CONFIDENTIAL

ACS-I Development Workbook

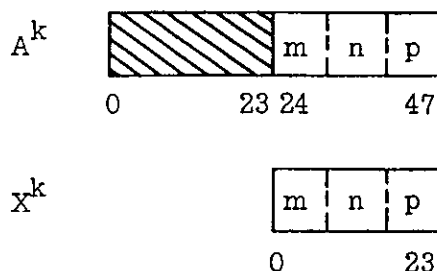
Page: 7-7

Date: 1/8/68

This page has been deleted.

Insert Field

Register R^k supplies three 8-bit integer parameters m , n , and p . These parameters are packed in R^k as shown:



The contents of register R^j are rotated left m positions. Bits rotated out of position 0 are inserted into position 47/23.

The $p-n$ bits of this rotated quantity numbered n , $n+1$, $n+2, \dots$, $p-1$ are then inserted into the corresponding bits of register R^i .

The remaining bits of R^i (namely those numbered $0, 1, 2, \dots, n-1$ and $p, p+1, p+2, \dots, 47/23$) either are left unaltered for the instructions IFX and IFA or are set to 0's for the instructions IFZX and IFZA. The contents of R^j and R^k are not changed.

The parameter m is interpreted as a positive integer modulo 48/24. The normal ranges for the positive integer parameters n and p are:

$$\begin{aligned} 0 &\leq n \leq 47/23 \\ 1 &\leq p \leq 48/24 \\ n &< p \end{aligned}$$

If $p \geq 49/25$, the operation proceeds as if $p = 48/24$. If $n \geq 48/24$ or if $p = 0$ or if $n = p$, the contents of R^i are left unaltered for IFA and IFX or are set to 0's for IFZA and IFZX.

If $n > p$, the $n - p$ bits of R^i numbered $p, p+1, \dots, n-1$ are set to 0's; the remaining bits are left unaltered for IFA and IFX or are also set to 0's for IFZA and IFZX.