

INSTRUCTION SET IN DIRECT MODE

Revision 4
Nov-02-2022

CODING

OPC	MNEMONIC	FLAGS			CODE											
					OPCODE				OPERAND X OPCODE				OPERAND Y			
1	ADD RX,RY	V	Z	C	0	0	0	1	X	X	X	X	Y	Y	Y	Y
2	ADC RX,RY	V	Z	C	0	0	1	0	X	X	X	X	Y	Y	Y	Y
3	SUB RX,RY	V	Z	C	0	0	1	1	X	X	X	X	Y	Y	Y	Y
4	SBB RX,RY	V	Z	C	0	1	0	0	X	X	X	X	Y	Y	Y	Y
5	OR RX,RY		Z		0	1	0	1	X	X	X	X	Y	Y	Y	Y
6	AND RX,RY		Z		0	1	1	0	X	X	X	X	Y	Y	Y	Y
7	XOR RX,RY		Z		0	1	1	1	X	X	X	X	Y	Y	Y	Y
8	MOV RX,RY				1	0	0	0	X	X	X	X	Y	Y	Y	Y
02	INC RY		Z	C	0	0	0	0	0	0	1	0	Y	Y	Y	Y
03	DEC RY		Z	C	0	0	0	0	0	0	1	1	Y	Y	Y	Y
0D	RRC RY		Z	C	0	0	0	0	1	1	0	1	Y	Y	Y	Y

ADD X,Y

Add register Y to register X

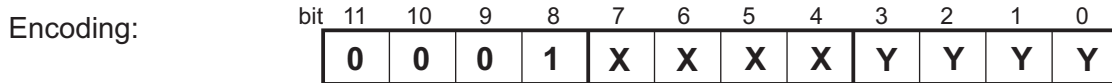
Syntax: {label} ADD X, Y

Operands: X ∈ #0...#15
Y ∈ #0...#15

Operation: $X \leftarrow X + Y$

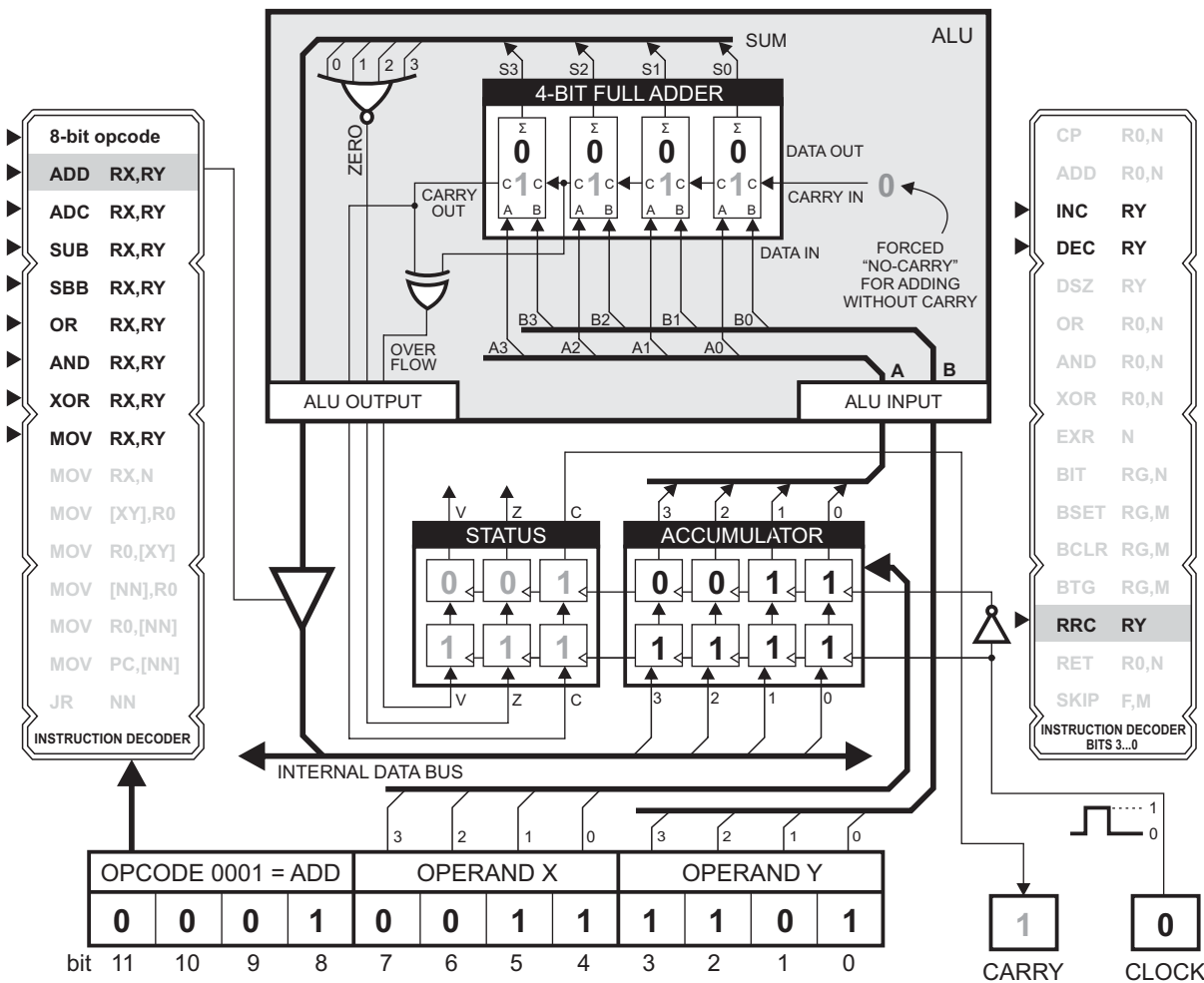
Description: Add with Carry contents of the register Y to the contents of the register X and place the result in the register X.

Flags affected: If there is the overflow (if $X + Y > 15$), set C.
Otherwise, reset C.
If result = 0000 after operation, set Z. Otherwise, reset Z.
If there is the underflow for signed representation, set V.



The "0001" bits are the ADD X,Y opcode
The "XXXX" bits are the contents of register X
The "YYYY" bits are the contents of register Y

Example: ADD 3, 13



ADC X,Y

Add with Carry register Y to register X

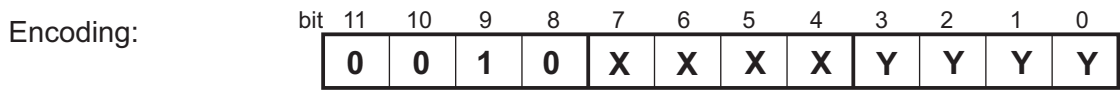
Syntax: {label} ADC X, Y

Operands: X ∈ #0...#15
Y ∈ #0...#15

Operation: $X \leftarrow X + Y + \text{Carry}$

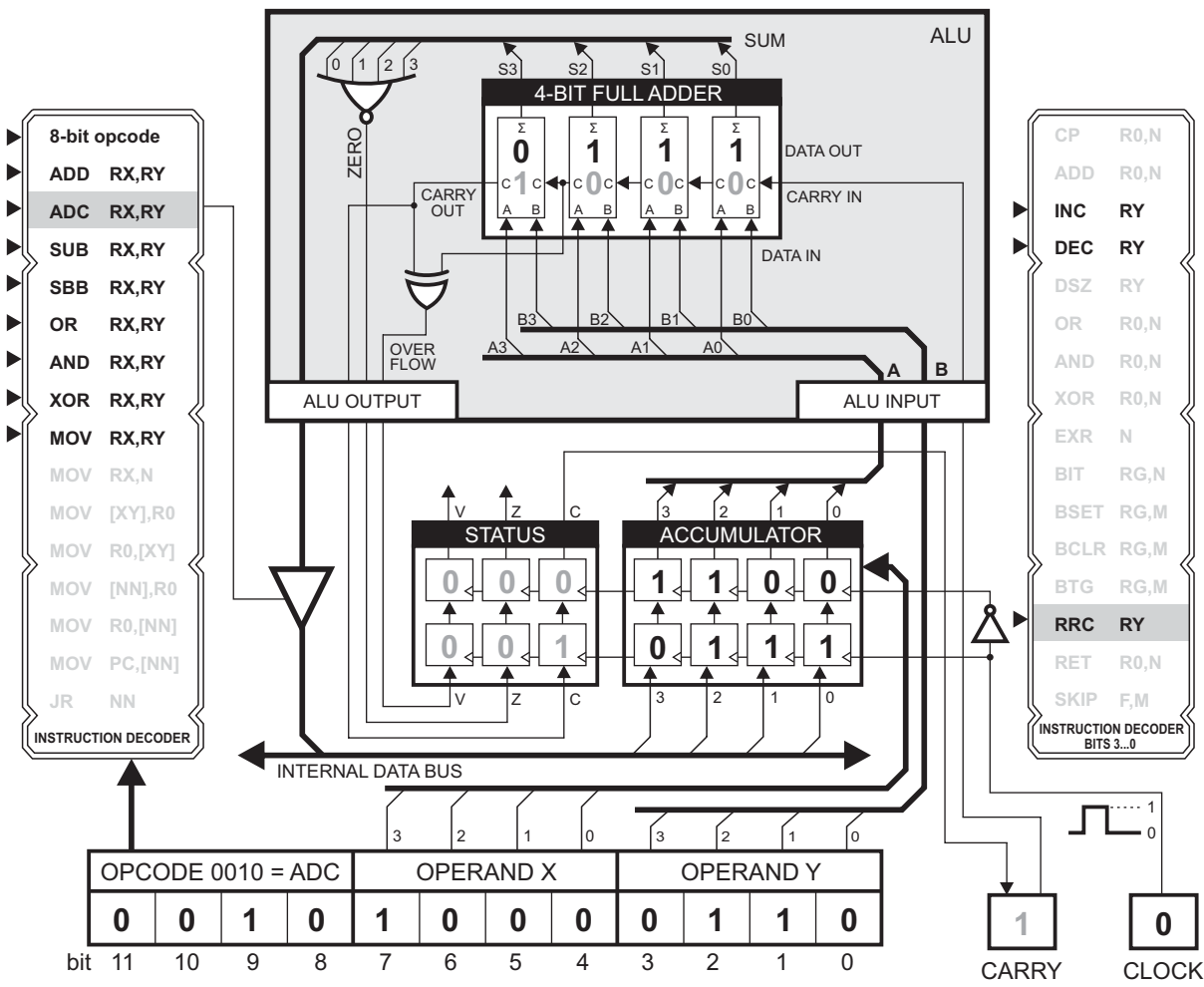
Description: Add with Carry contents of the register Y to the contents of the register X and place the result in the register X.

Flags affected: If there is the overflow (if $X + Y + \text{Carry} > 15$), set C. Otherwise, reset C.
If result = 0000 after operation, set Z. Otherwise, reset Z.
If there is the underflow for signed representation, set V.



The "0010" bits are the ADD X,Y opcode
The "XXXX" bits are the contents of register X
The "YYYY" bits are the contents of register Y

Example: ADC 8, 6 (Carry set)



SUB X,Y

Subtract register Y from register X

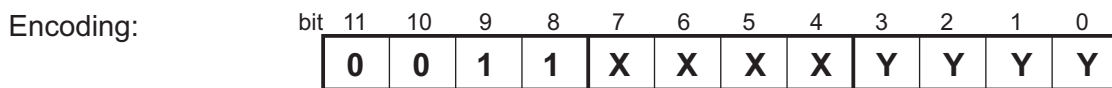
Syntax: {label} SUB X, Y

Operands: X ∈ #0...#15
Y ∈ #0...#15

Operation: $X \leftarrow X - Y$

Description: Subtract the contents of the register Y from the contents of the register X and place the result in the register X.

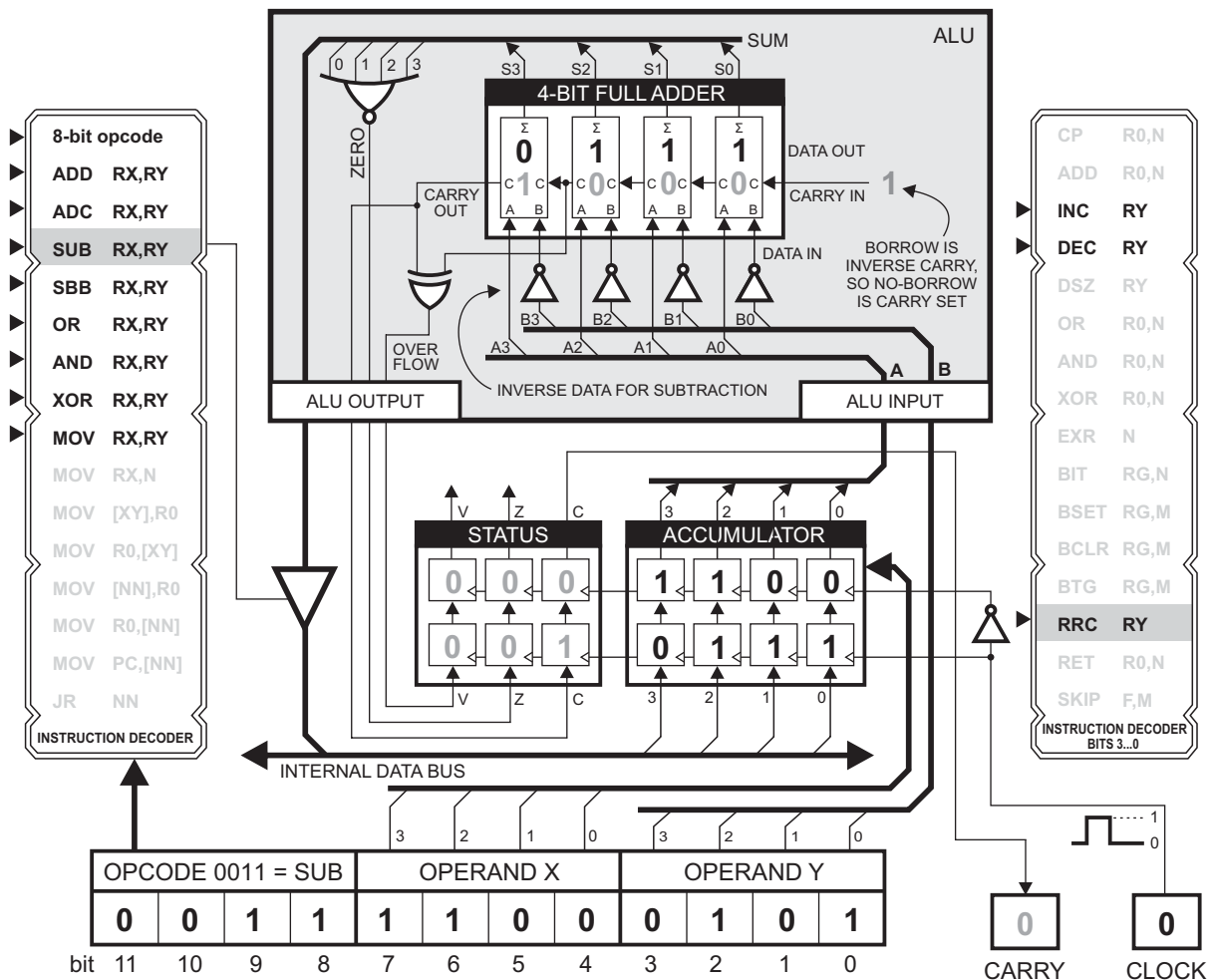
Flags affected: If there is the underflow (if $Y < X$), reset C.
Otherwise, set C. (note: Borrow is inverse C)
If result = 0000 after operation, set Z. Otherwise, reset Z.
If there is the underflow for signed representation, set V.



The "0011" bits are the SUB X,Y opcode
The "XXXX" bits are the contents of register X
The "YYYY" bits are the contents of register Y

Example:

SUB 12,5



SBB X,Y

Subtract register Y from register X with borrow

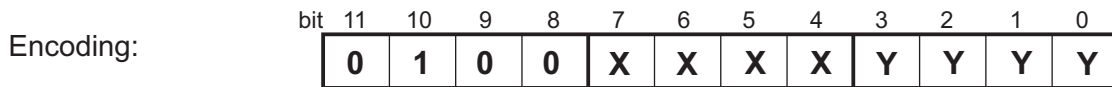
Syntax: {label} SBB X, Y

Operands: X ∈ #0...#15
Y ∈ #0...#15

Operation: $X \leftarrow X - Y - \overline{\text{Carry}}$

Description: Subtract the contents of the register Y and inverse Carry flag from the contents of the register X and place the result in the register X.

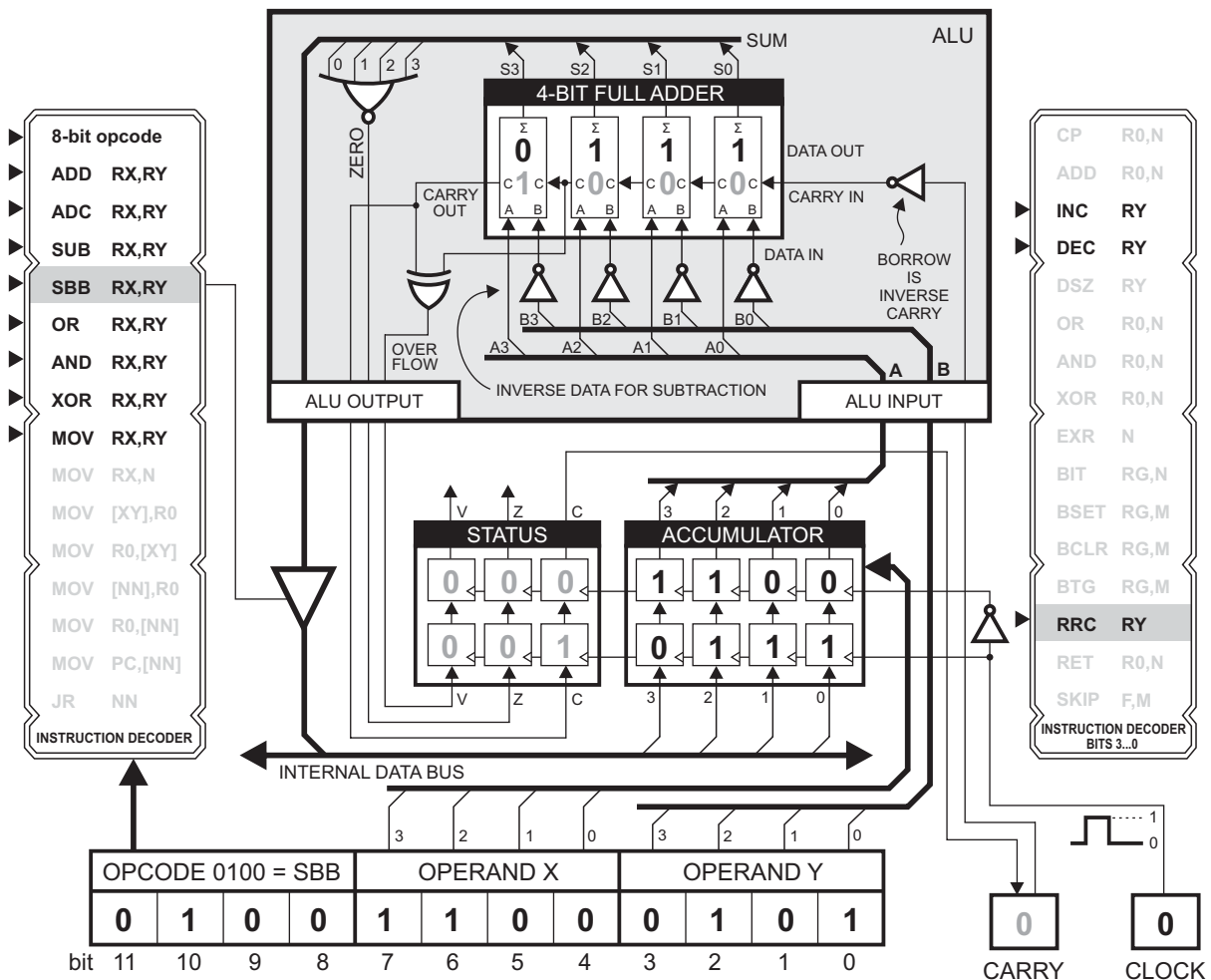
Flags affected: If there is the underflow (if $Y < X$), reset C.
Otherwise, set C. (note: Borrow is inverse C)
If result = 0000 after operation, set Z. Otherwise, reset Z.
If there is the underflow for signed representation, set V.



The "0100" bits are the Sbb X,Y opcode
The "XXXX" bits are the contents of register X
The "YYYY" bits are the contents of register Y

Example:

SBB 12, 5



OR X,Y

Inclusive OR registers X and Y

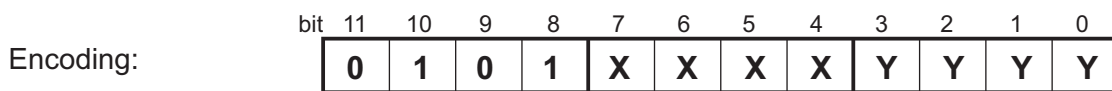
Syntax: {label} OR X, Y

Operands: X ∈ #0...#15
Y ∈ #0...#15

Operation: $X \leftarrow X .OR. Y$

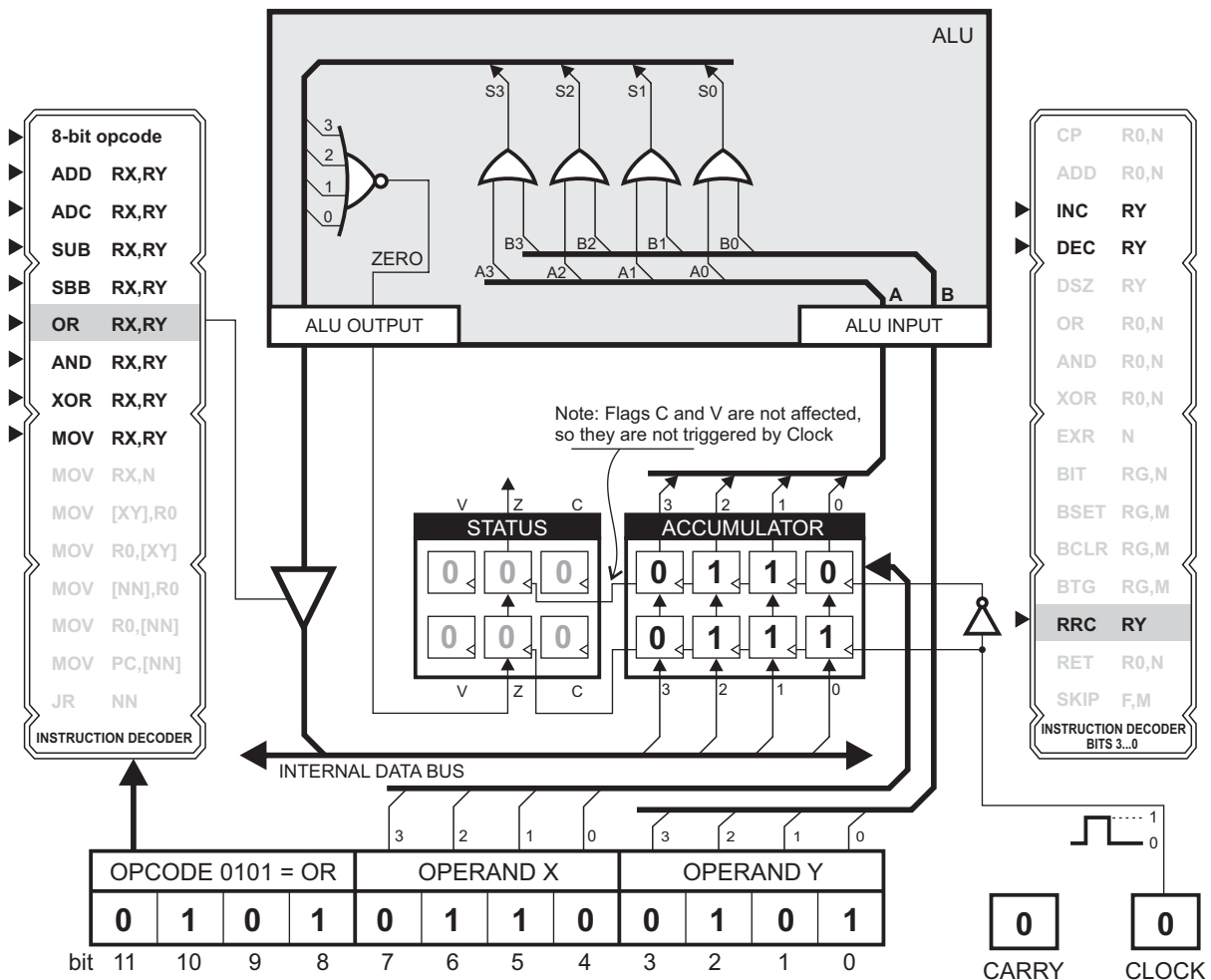
Description: Compute the logical inclusive OR operation of register X and register Y and place the result into the register X.

Flags affected: Flag C is not affected.
If result = 0000 after operation, set Z. Otherwise, reset Z.



The "0101" bits are the OR X,Y opcode
The "XXXX" bits are the contents of register X
The "YYYY" bits are the contents of register Y

Example: OR 6,5



AND X,Y

Logical AND registers X and Y

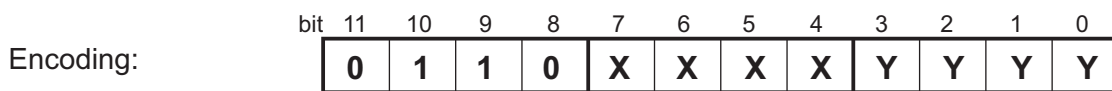
Syntax: {label} AND X, Y

Operands: X ∈ #0...#15
Y ∈ #0...#15

Operation: X ← X .AND. Y

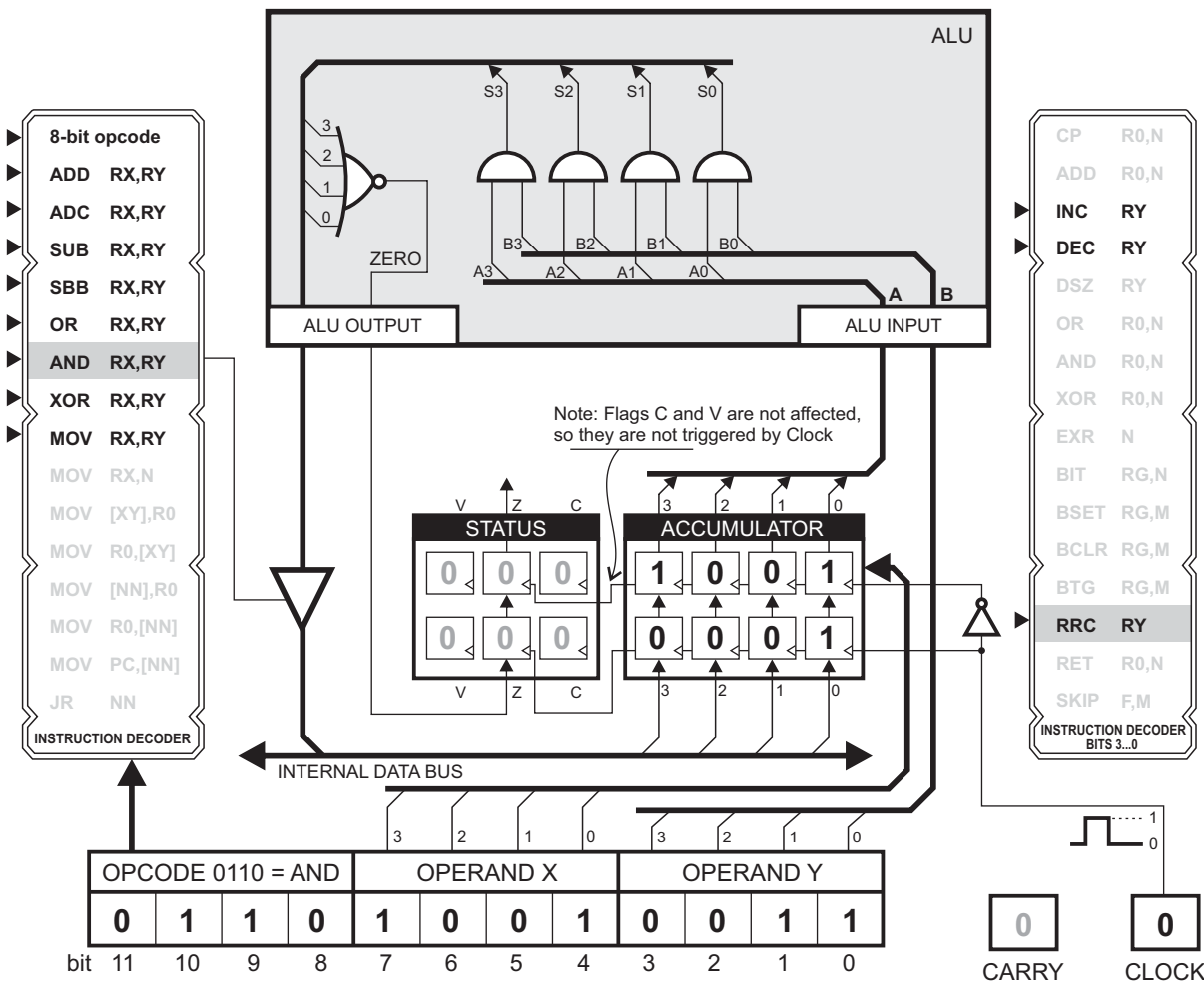
Description: Compute the logical inclusive OR operation of register X and register Y and place the result into the register X.

Flags affected: Flag C is not affected.
If result = 0000 after operation, set Z. Otherwise, reset Z.



The "0110" bits are the AND X,Y opcode
 The "XXXX" bits are the contents of register X
 The "YYYY" bits are the contents of register Y

Example: AND 9, 3



XOR X,Y

Exclusive OR registers X and Y

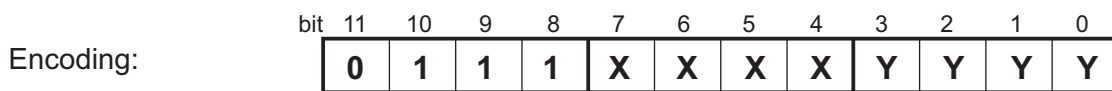
Syntax: {label} XOR X, Y

Operands: X ∈ #0...#15
Y ∈ #0...#15

Operation: X ← X .OR. Y

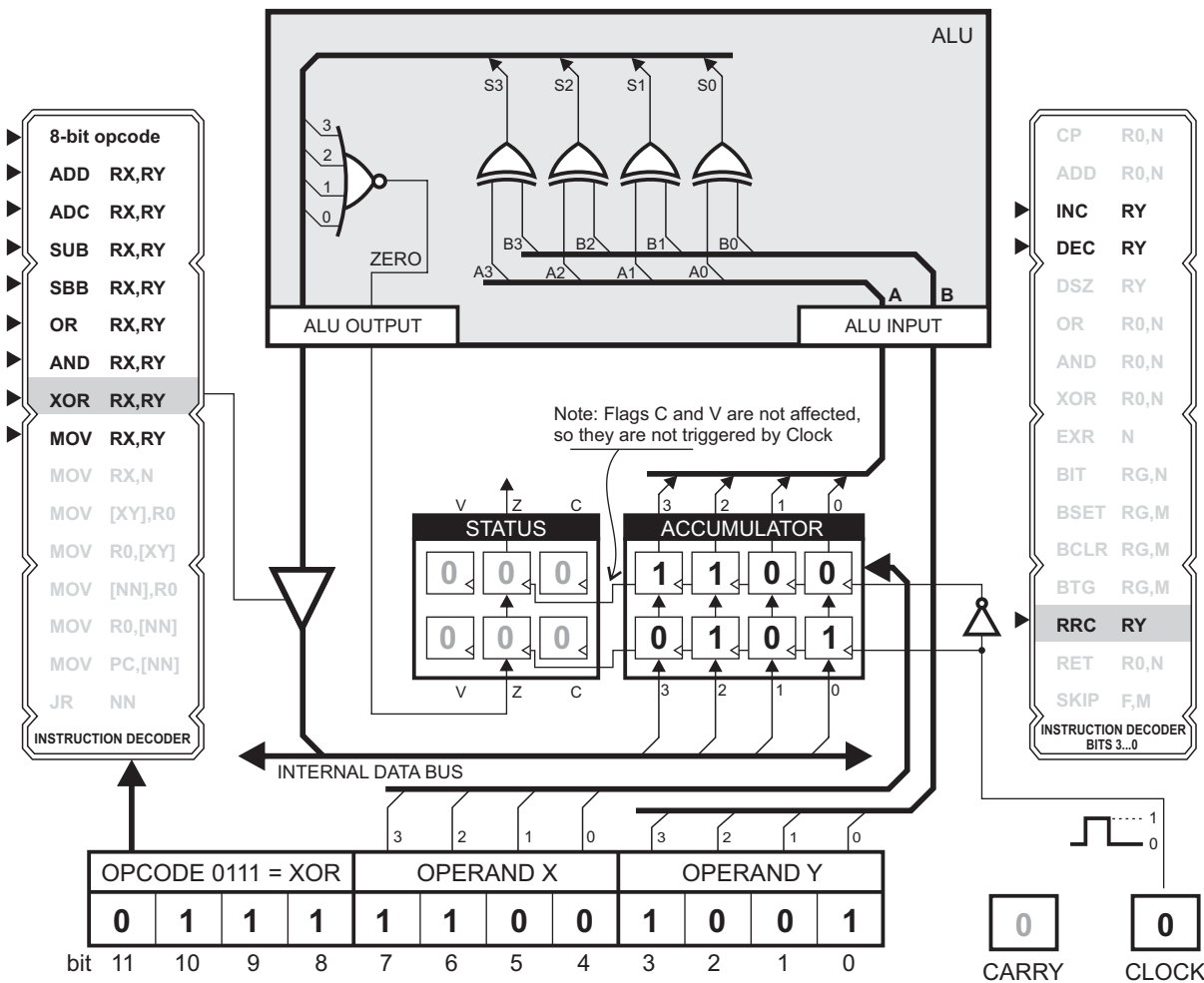
Description: Compute the logical exclusive OR operation of register X and register Y and place the result into the register X.

Flags affected: Flag C is not affected.
If result = 0000 after operation, set Z. Otherwise, reset Z.



The "0111" bits are the XOR X,Y opcode
 The "XXXX" bits are the contents of register X
 The "YYYY" bits are the contents of register Y

Example: XOR 12, 9



MOV X,Y

Move contents of register Y to register X

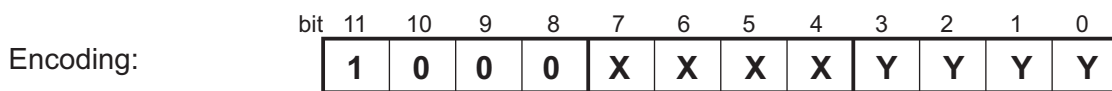
Syntax: {label} MOV X, Y

Operands: X ∈ #0...#15
Y ∈ #0...#15

Operation: X ← Y

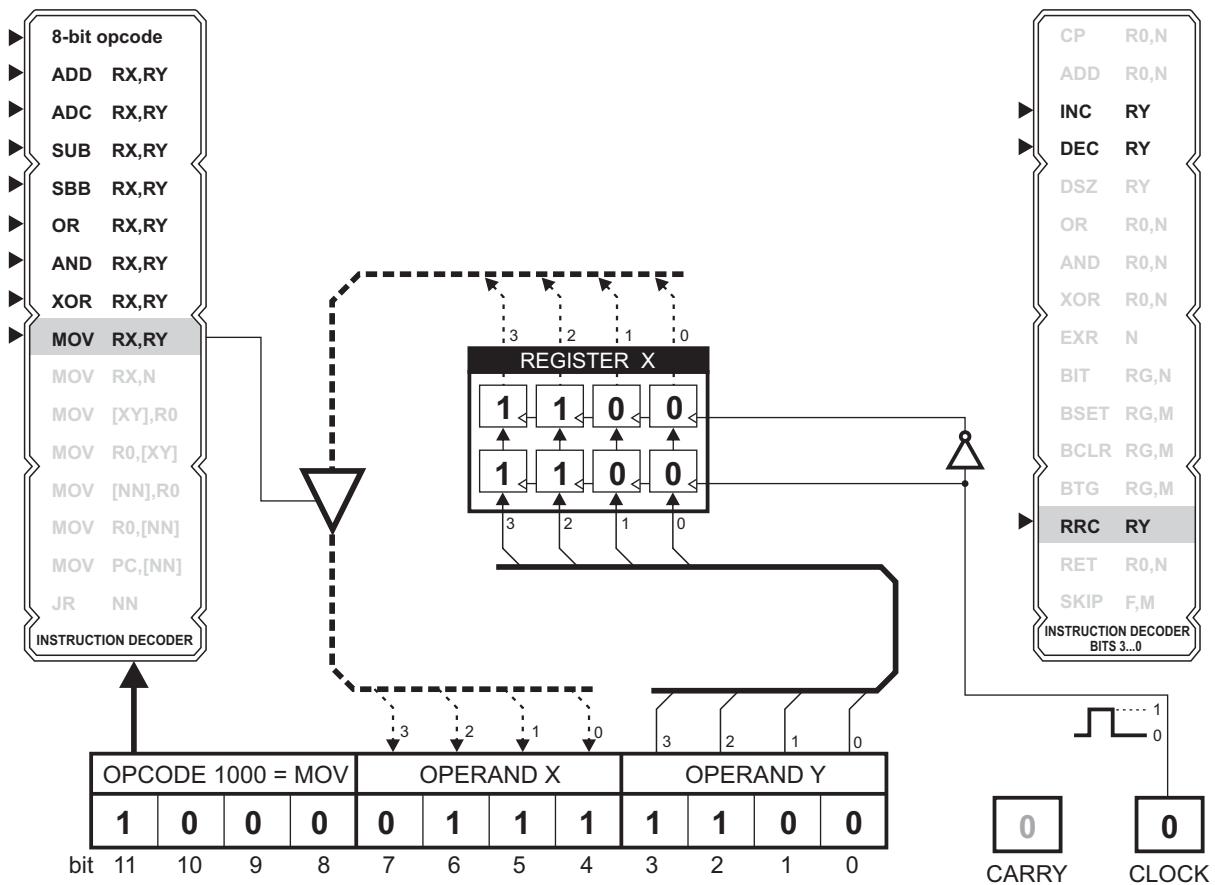
Description: Move the contents of the register Y to register X. Value of the register Y is unchanged.

Flags affected: None.



The "1000" bits are the MOV X,Y opcode
 The "XXXX" bits are the old contents of register X
 The "YYYY" bits are the contents of register Y

Example: MOV 7, 12



INC Y

Increment the value of register Y

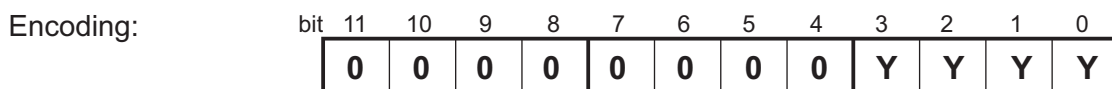
Syntax: {label} INC Y

Operands: $Y \in \#0\ldots\#15$

Operation: $Y \leftarrow Y + 1$

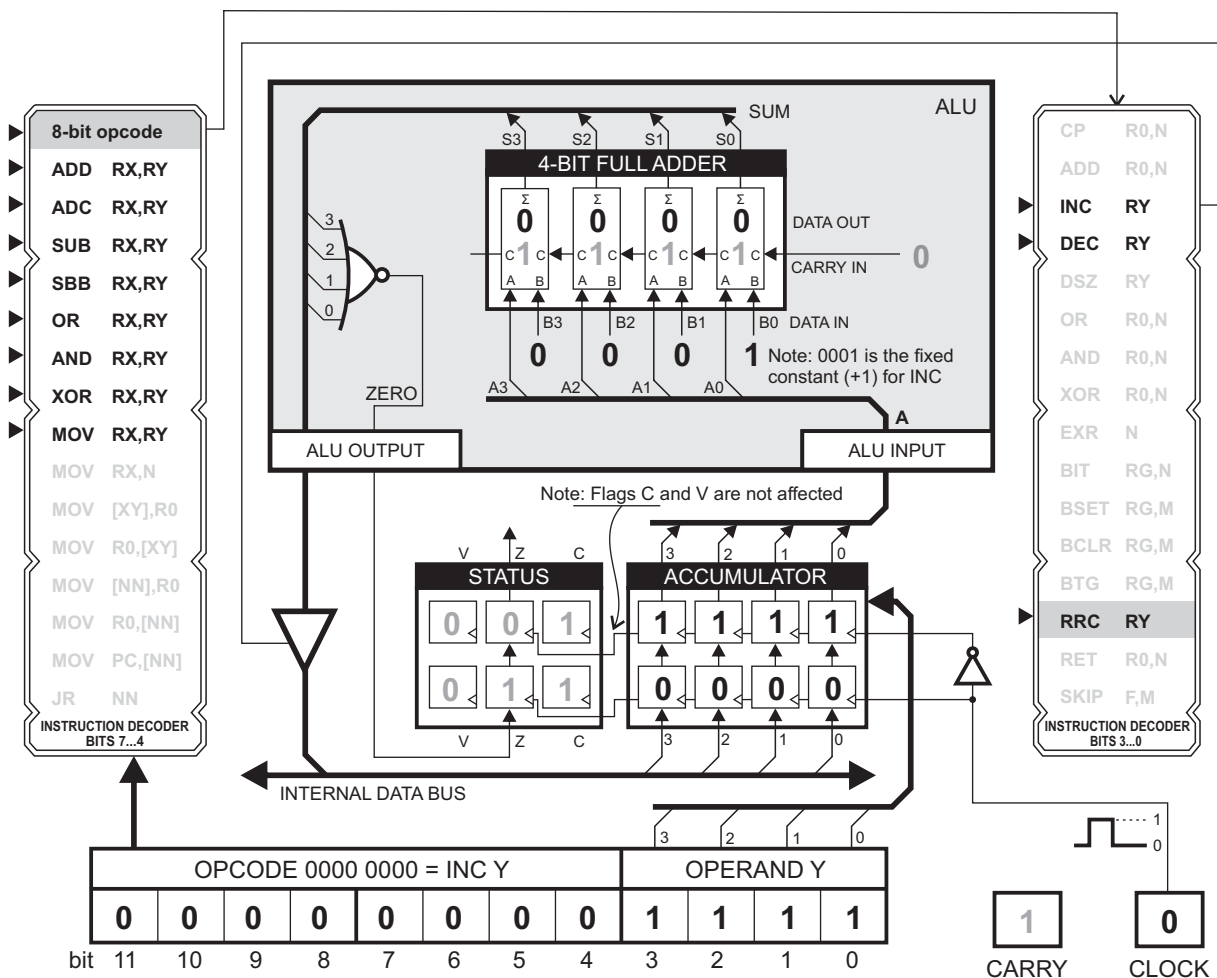
Description: Add 1 to the contents of the 4-bit register Y and place the result back into the register Y.

Flags affected: Flag C is not affected.
If result = 0000 after operation, set Z. Otherwise, reset Z.



The "0000 0000" bits are the INC Y opcode
The "YYYY" bits are the contents of register Y

Example: INC 15



DEC Y

Decrement the value of register Y

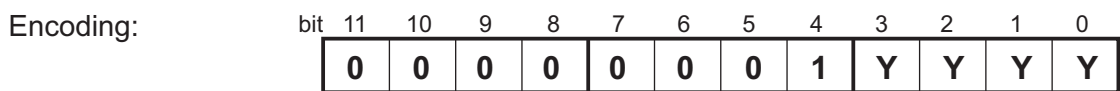
Syntax: {label} DEC Y

Operands: $Y \in \#0\#\dots\#15$

Operation: $Y \leftarrow Y - 1$

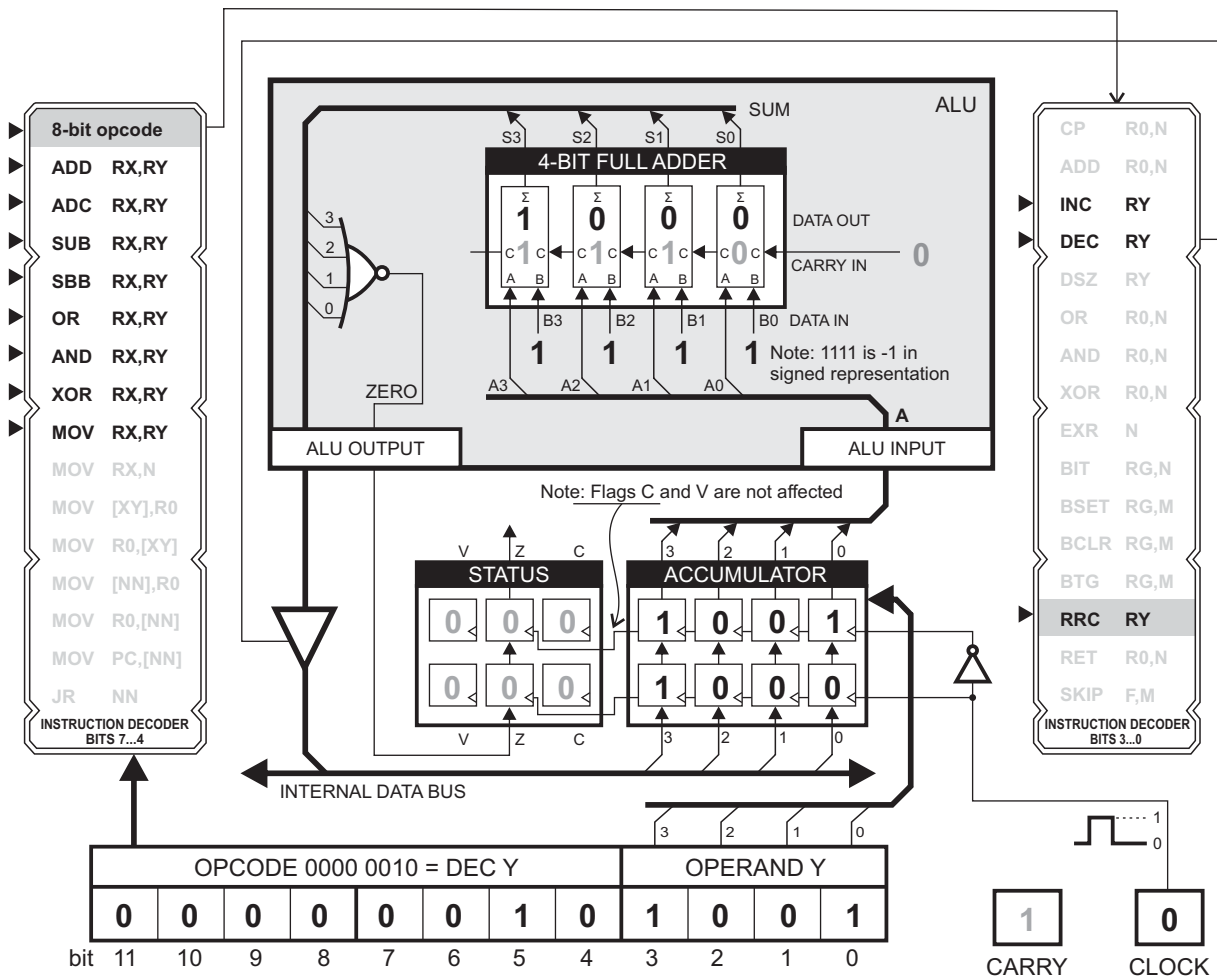
Description: Add -1 to the contents of the 4-bit register Y and place the result back into the register Y.

Flags affected: Flag C is not affected.
If result = 0000 after operation, set Z. Otherwise, reset Z.



The "0000 0001" bits are the DEC Y opcode
The "YYYY" bits are the contents of register Y

Example: DEC 9



RRC Y

Rotate right through Carry the value of register Y

Syntax: {label} RRC Y

Operands: $Y \in \#0\ldots\#15$

Operation: $C \leftarrow Y_0, Y_3 \leftarrow C, Y_2 \leftarrow Y_3, Y_1 \leftarrow Y_2, Y_0 \leftarrow Y_1$

Description: Rotate the contents of the register Y one bit to the right through Carry and place the result back in the register Y. The Carry flag is shifted into the Bit 7 of register Y, and Carry is overwritten with the Bit 0 of register Y.

Flags affected: Flag C is not affected.
If result = 0000 after operation, set Z. Otherwise, reset Z.

Encoding:

bit	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	1	0	Y	Y	Y	Y

The "0000 0010" bits are the RRC Y opcode
The "YYYY" bits are the contents of register Y

Example: RRC 6

