SET(Latch) & RESET(unlatch)

*Set syntax:* SET address(W0.00)

*RESET syntax:* RSET address(W0.00)

***Functional Block Diagram***

Graphical language for

Programmable logic controller

This defines the function b/w

i/p & o/p variables.

***if (condition) then***

***:=statement;***

***end\_if;***

**if else**

**if(condition) then**

**:=Statement1;**

**Else**

**:=statement2;**

**End\_if;**

**Boolean-> 0/1 True/flase On/Off**

**Integer-> 16bit memory /1 word**

**int -> -32768 to +32767**

**uint-> 0 to 65535**

**dint-> double integer(32bit memory)**

**dint-> -2147483648 to +2147483647**

**udint-> o to 42949667295**

**TASK:**

**SW1 🡪 ON**

**Motor 🡪 ON**

**SW2 🡪 ON**

**Motor 🡪 OFF**

**IF SWITCH1 = TRUE THEN**

**MOTOR := TRUE;**

**END\_IF;**

**IF SWITCH2 = TRUE THEN**

**MOTOR := FALSE;**

**END\_IF;**

***IF & ELSIF***

IF CONDITION THEN

:=Statement1;

ELSIF CONDITION THEN   
 := Statement2;

ELSE

:= Statement3;

END\_IF;

IF VALUE = 1 THEN

MOTOR3 := TRUE;

MOTOR4 := FALSE;

ELSIF VALUE = 2 THEN

MOTOR3 := FALSE;

MOTOR4 := TRUE;

ELSE

MOTOR3 := FALSE;

MOTOR4 := FALSE;

END\_IF;

***DATA REGISTER***

Data register address: D0,D1,D2……..D3999

🡪Single data register carry 16bit of data memory.

***MOVE OPERATION***

SYNTEX: MOV S D where S-> Source, D->Destination

XFER &5 S D &5 defines no.of data is to be moved

***ARITHMETIC OPERATION:***(INTEGER)

ADD: + D20 D21 D22 //D20+D21=D22

SUB: - D24 D25 D26 //D24-D25=D26

Note: The result value should not exceed the integer

Range values.

MUL: \* D27 D28 D29 //D27\*D28=D29

DIV: / D30 D31 D32 // D30/D31=D32

Where D32 stores quotient

(next successive address of OUTPUT) 🡪

D33 stores Remainder