CMR

INSTITUTE OF

TECHNOLOGY

USN | | | | | | | | | | |

Internal Assesment Test - III

Sub:	Programmable Logic Conrtroller					Code:	15EE562
Date:	22/11/2018	V	Sem	Max Marks:	50	Duration:	90 minutes

Answer Any FIVE FULL Questions

		Marks	OBE	
			CO	RBT
1	Explain data comparison & arithmetic operations instructions with ladder diagram.	[10]	CO5	L4
2	Describe the operation of data manipulation with an example.	[10]	CO5	L2
3	Explain the implementation of closed loop control system using PLC.	[10]	CO6	L4
4	Explain different methods by which the controller can react to an error signal in PLC closed loop control schemes with neat block diagram.	[10]	CO6	L4
5	Write a ladder diagram for a vessel filling condition using any relevant math instruction. This application requires an alarm sound for the following condition, When a supply system leaks 5 lb or more of raw material into the vessel, after a preset weight of 500 lb has been reached.	[10]	CO5	L4
6	Write a ladder diagram for converting Celsius temperature to Fahrenheit & explain the operation.	[10]	CO5	L3
7	Explain the operations of master control relay.	[10]	CO6	L4
8	Explain with ladder diagram and the sequence signals of a bit shift register.	[10]	CO6	L4

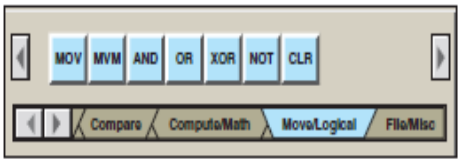


Figure 2

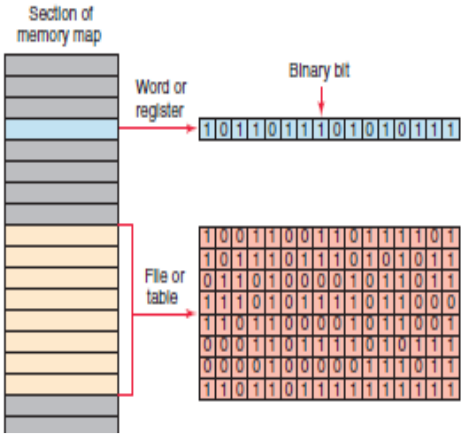


Figure 1

2. Data manipulation instructions allow numerical data stored in the controller's memory to be operated on within the control program.

The use of data manipulation extends a controller's capability from that of simple on/off control based on binary logic, to quantitative decision making involving data comparisons, arithmetic, and conversions—which in turn can be applied to analog and positioning control.

There are two basic classes of instructions to accomplish data manipulation: instructions that operate on word data and those that operate on file, or block, data, which involve multiple words.
Each data manipulation instruction requires two or more words of data memory for operation. The words of data memory in singular form may be referred to either as *registers* or as *words,* depending on the manufacturer.
The terms *table* or *file* are generally used when a *consecutive* group of related data memory words is referenced.
Figure 1 illustrates the difference between a word and a file. The data contained in files and words will be in the form of binary *bits* represented as series of 1s and 0s.
The data manipulation instructions allow the movement, manipulation, or storage of data in either single- or multiple-word groups from one data memory area of the PLC to another. Use of these PLC instructions in applications that require the generation and manipulation of large quantities of data greatly reduces the complexity and quantity of the programming required.

Figure -2 shows the **Move/Logical** menu tab for the SLC 500 PLC and its associated RSLogix software. The commands can be summarized as follows:
MOV (Move) —Moves the source value to the destination.
MVM (Masked Move) —Moves data from a source location to a selected portion of the destination.
AND (And) —Performs a bitwise AND operation.
OR (Or) —Performs a bitwise OR operation.
XOR (Exclusive Or) —Performs a bitwise XOR operation.
NOT (Not) —Performs a bitwise NOT operation.
CLR (Clear) —Sets all bits of a word to zero.

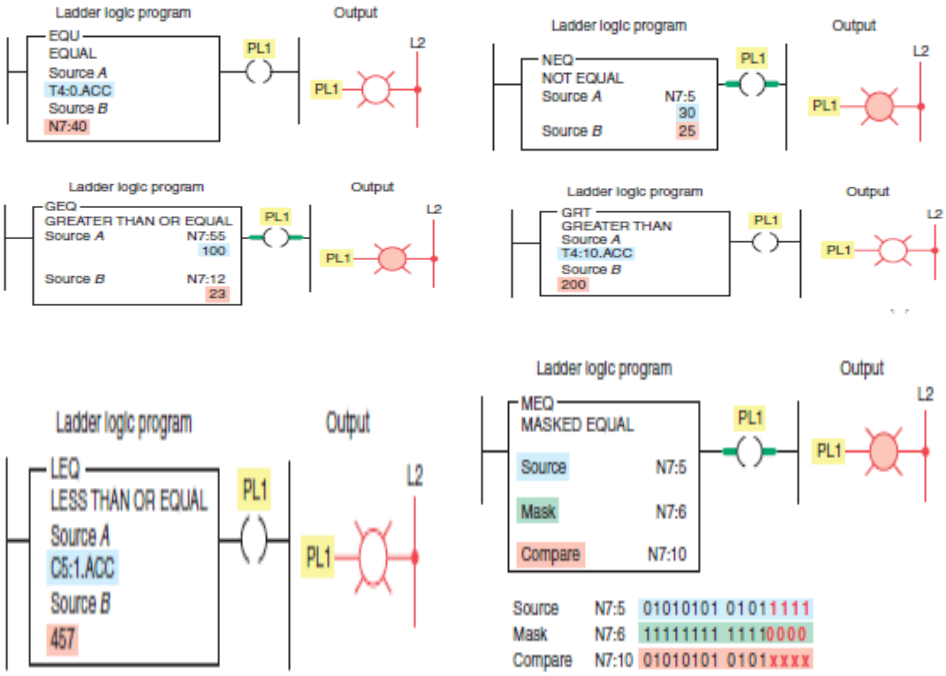
1. Data Compare Instructions

Data transfer operations are all output instructions, whereas *data compare* instructions are *input* instructions.

Data compare instructions are used to compare numerical values. These instructions compare the data stored in two or more words (or registers) and make decisions based on the program instructions. Numeric values in two words of memory can be compared for each of the basic data compare instructions shown in Figure , depending on the PLC.

Name	Symbol
Equal to	(=)
Not equal to	(≠)
Less than	(<)
Greater than	(>)
Less than or equal to	(≤)
Greater than or equal to	(≥)

The compare instructions can be summarized as follows:
LIM (Limit test) —Tests whether one value is within the limit range of two other values.
MEQ (Masked Comparison for Equal) —Tests portions of two values to see whether they are equal. Compares 16-bit data of a source address to 16-bit data at a reference address through a mask.
EQU (Equal) —Tests whether two values are equal.
NEQ (Not Equal) —Tests whether one value is not equal to a second value.
LES (Less Than) —Tests whether one value is less than a second value.
GRT (Greater Than) —Tests whether one value is greater than a second value.
LEQ (Less Than or Equal) —Tests whether one value is less than or equal to a second value.
GEQ (Greater Than or Equal) —Tests whether one value is greater than or equal to a second value.



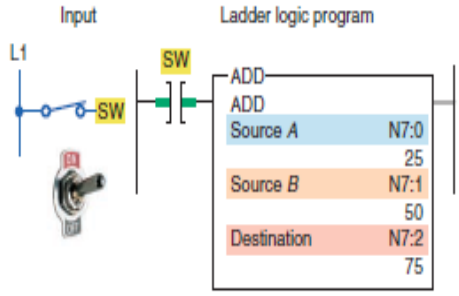
2. The basic four mathematical functions performed by PLCs are:

Addition —The capability to add one piece of data to another.
Subtraction —The capability to subtract one piece of data from another.
Multiplication —The capability to multiply one piece of data by another.
Division —The capability to divide one piece of data by another.

Math instructions use the contents of two words or registers and perform the desired function. The PLC instructions for data manipulation (data transfer and data compare) are used with the math symbols to perform math functions.

Addition Instruction

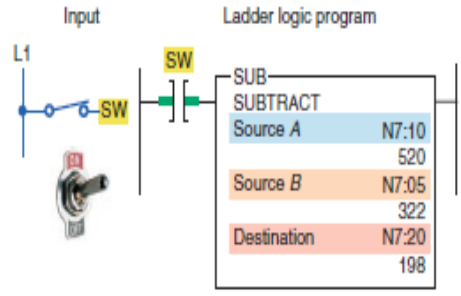
Most math instructions take two input values, perform the specified arithmetic function, and output the result to an assigned memory location. the *ADD* instruction performs the addition of two values stored in the referenced memory locations.



Subtraction Instruction

The *SUB (subtract)* instruction is an output instruction that subtracts one value from another and stores the result in the destination address. When rung conditions are true, the subtract instruction subtracts source *B* from source *A* and stores the result in the destination. Figure shows the SUB instruction used with the SLC 500 controllers. The operation of the logic rung can be summarized as follows:

• When input switch SW is closed the rung will be true.
• The value stored at the source *B* address, N7:05 (322), is subtracted from the value stored at the source *A* address, N7:10 (520).
• The answer (198) is stored at the destination address, N7:20.
• Source *A* and source *B* can be either values or addresses that contain values, but *A* and *B* cannot both be constants.

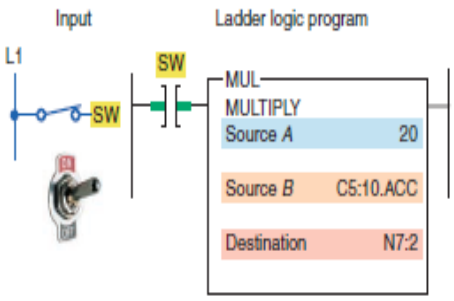


Multiplication Instruction

The *multiply (MUL)* instruction is an output instruction that multiplies two values and stores the result in the destination address. Figure shows the MUL instruction used with the SLC 500 controllers. The operation of the logic rung can be summarized as follows:

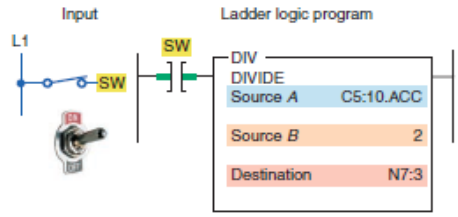
• When input switch SW is closed the rung will be true.

The data in source *A* (constant 20) will be multiplied by the data in source *B* (accumulated value ofcounter C5:10).
• The resultant answer is placed in the destination N7:2.



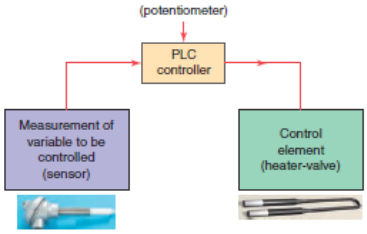
Division Instruction

The *divide (DIV)* instruction divides the value in source *A* by the value in source *B* and stores the result in the destination and math register. Figure shows an example of the DIV instruction. The operation of the logic rung can be summarized as follows:
• When input switch SW is closed the rung will be true.
• The data in source *A* (the accumulated value of counter C5:10) is then divided by the data in source *B* (the constant 2).
• The result is placed in the destination N7:3.
• If the remainder is 0.5 or greater, a roundup occurs in the integer destination.
• The value stored in the math register consists of the unrounded quotient (placed in the most significant word) and the remainder (placed in the least significant word).



3.

- A closed-loop system utilizes feedback to measure the actual system operating parameter being controlled such as temperature, pressure, flow, level, or speed. This feedback signal is sent back to the PLC where it is compared with the desired system set-point. The controller develops an error signal that initiates corrective action and drives the final output device to the desired value.
- When a control system is designed such that it receives operating information from the machine and makes adjustments to the machine based on this operating information, the system is said to be a closed-loop system.
- The block diagram of a closed-loop control system is shown in Figure . A measurement is made of the variable to be controlled. This measurement is then compared to a reference point, or set-point. If a difference (error) exists between the actual and desired levels, the PLC control program will take the necessary corrective action. Adjustments are made continuously by the PLC until the difference between the desired and actual output is as small as is practical.
- With on/off PLC control (also known as *two-position* and *bang-bang control*), the output or final control element is either on or off—one for the occasion when the value of the measured variable is above the set-point and the other for the occasion when the value is below the set-point. The controller will never keep the final control element in an intermediate position. Most residential thermostats are on/off type controllers.
- On/off control is inexpensive but not accurate enough for most process and machine control applications. On/ off control almost always means overshoot and resultant system cycling. For this reason a *deadband* usually exists around the set-point. The deadband or hysteresis of the control loop is the difference between the on and off operating points.
- *Proportional controls* are designed to eliminate the hunting or cycling associated with on/off control. They allow the final control element to take intermediate positions between on and off. This permits *analog control* of the final control element to vary the amount of energy to the process, depending on how much the value of the measured variable has shifted from the desired value.

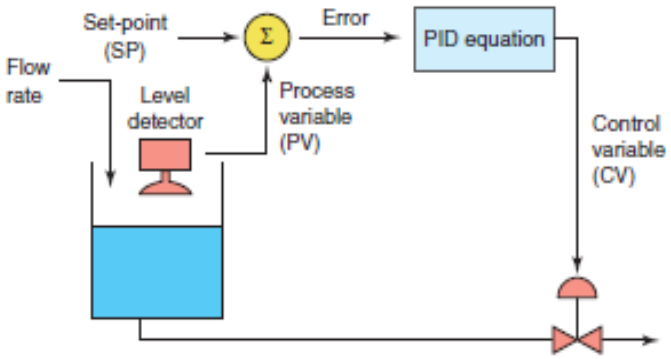


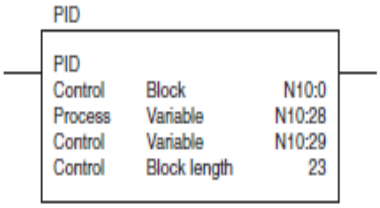
4. Typical PID control loop is illustrated in Figure below. The loop measures the process, compares it to a set-point, and then manipulates the output in the direction which should move the process toward the set-point. The terminology used in conjunction with a PID loop can be summarized as follows:

- Operating information that the controller receives from the machine is called the *process variable (PV)* or *feedback.*
- • Input from the operator that tells the controller the desired operating point is called the *set-point (SP).*
- • When operating, the controller determines whether the machine needs adjustment by comparing (by subtraction) the set-point and the process variable to produce a difference (the difference is called the *error*).

• Output from the loop is called the *control variable (CV),* which is connected to the controlling part of the process.

• The PID loop takes appropriate action to modify the process operating point until the control variable and the set-point are very nearly equal.





PID instruction parameters can be summarized as follows:

• Control Block is the file that stores the data required to operate the instruction.

• Process Variable (PV) is an element address that stores the process input value.

• Control Variable (CV) is an element address that stores the output of the PID instruction.

5. The operation of the program can be summarized as follows:

• When the start button is pressed, the fill solenoid (rung 1) and filling indicating light (rung 2) are turned on and raw material is allowed to flow into the vessel.

• The vessel has its weight monitored continuously by the PLC program (rung 3) as it fills.

• When the weight reaches 500 lb, the fill solenoid is de-energized and the flow is cut off.

• At the same time, the filling pilot light indicator is turned off and the full pilot light indicator (rung 3) is turned on.

• Should the fill solenoid leak 5 lb or more of raw material into the vessel, the alarm (rung 5) will energize and stay energized until the overflow level is reduced below the 5-lb overflow limit.

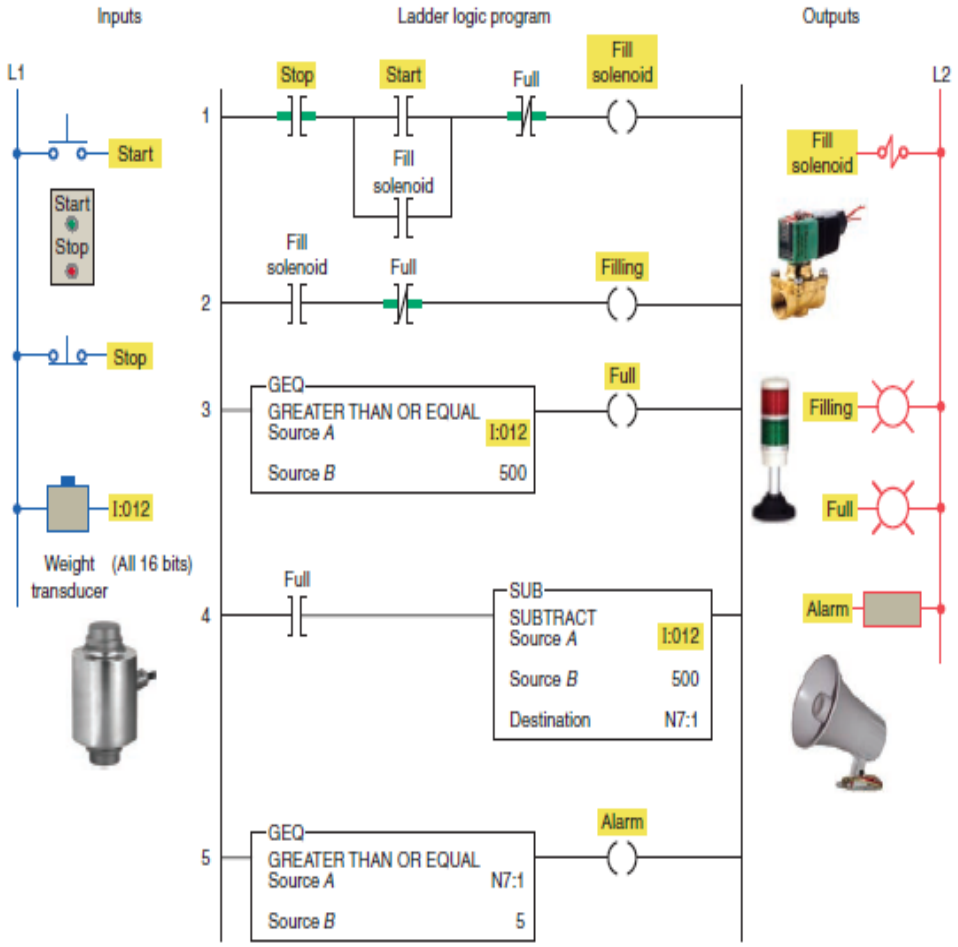


Figure 11-7 Vessel overfill alarm program.

6. The operation of the program can be summarized as follows:

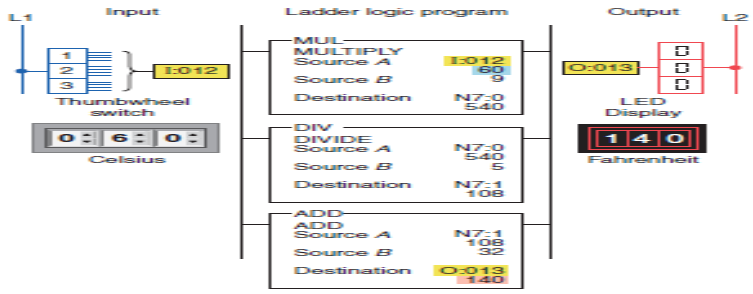
• The thumbwheel switch connected to the input module indicates Celsius temperature.

• The program is designed to convert the recorded Celsius temperature in the data table to Fahrenheit Values for display.

• Next, the DIV instruction divides 5 into the 540 and stores the answer (108) in address N7:1.

• Finally, the ADD instruction adds 32 to the value of 108 and stores the sum (140) in address O:13.

• Thus 60°C 5 140°F.



8. Bit Shift Registers

A bit *shift register* is a register that allows the shifting of bits through a single register or group of registers.
The bit shift register shifts bits serially (from bit to bit) through an array in an orderly fashion.
A shift register can be used to simulate the movement, or *track* the flow, of parts and information.
Common applications for shift registers include the following:

- Tracking of parts through an assembly line
- Controlling of machine or process operations
- Inventory control
- System diagnostics

Figure-1 illustrates the basic concept of a shift register. A shift pulse or clock causes each bit in the shift register to move 1 position to the right. At some point, the number of data bits fed into the shift register will exceed the register's storage capacity. When this happens, the first data bits fed into the shift register by the shift pulse are lost at the end of the shift register. Typically, data in the shift register could represent the following:

- Part types, quality, and size
- The presence or absence of parts
- The order in which events occur
- Identification numbers or locations
- A fault condition that caused a shutdown

A shift register to shift status data either right or left, as illustrated in Figure 2 , by shifting either status or values through data files. When you want to track parts on a status basis, use bit shift registers. Bit shift instructions will shift bit status from a source bit address, through a data file, and out to an unload bit, one bit at a time. There are two bit shift instructions:
bit shift left (BSL), which shifts bit status from a lower address number to a higher address number through a data file, and
bit shift right (BSR), which shifts data from a higher address number to a lower address number through a data file.

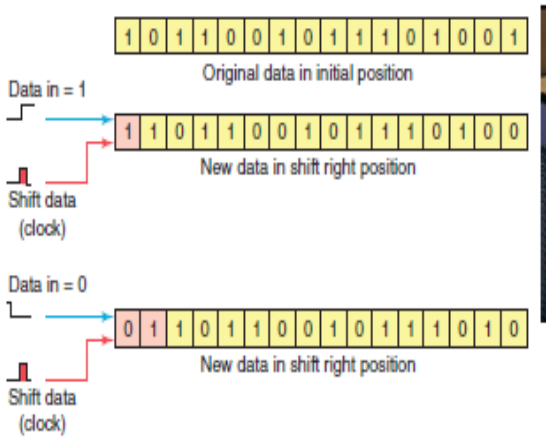


Figure-1

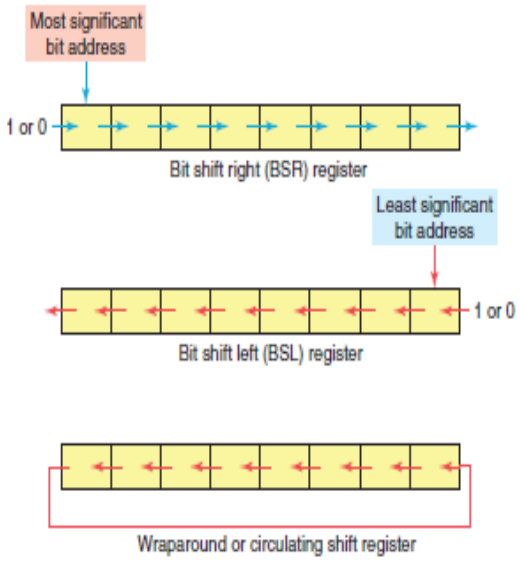


Figure -2

Some PLCs provide a *circulating shift register* function, which allows you to repeat a pattern again and again. When working with a bit shift register, you can identify each bit by its position in the register. Therefore, working with any bit in the register becomes a matter of identifying the position it occupies rather than the conventional word number/bit number addressing
scheme.
Figure 3 shows the **File Shift** menu tab and BSL and BSR instruction blocks that are part of the instruction set for the Allen-Bradley SLC 500 controllers. The commands can be summarized as follows:
BSL (Bit Shift Left) —Loads a bit of data into a bit array, shifts the pattern of data through the array to the left, and unloads the last bit of data in the array.
BSR (Bit Shift Right) —Loads a bit of data into a bit array, shifts the pattern of data through the array to the right, and unloads the last bit of data in the array.

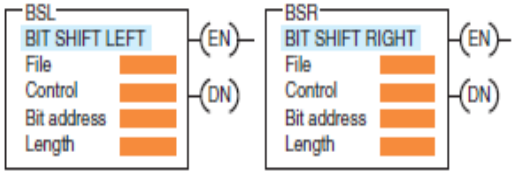


Figure-3

7. A *Master Control Reset (MCR)* instruction is an output coil instruction that functions like a master controlrelay. MCR coil instructions are used in pairs and can be programmed to control an entire circuit or to control only selected rungs of a circuit.
In the program , the MCR is programmed to control an entire circuit. The operation of the program can be summarized as follows:
• When the MCR instruction is false, or de-energized, all nonretentive (nonlatched) rungs below the MCR will be de-energized even if the programmed logic for each rung is true.
• All *retentive* rungs will remain in their *last state*.
• The MCR instruction establishes a zone in the user program in which all non-retentive outputs can be turned off simultaneously.
• *Retentive* instructions should not normally be placed within an MCR zone because the MCR zone maintains retentive instructions in the last active state when the instruction goes false.
• An off-delay timer will start timing when in a deenergized MCR zone.