1.12 Binary Logic Diagrams*

G. PLATT (1970, 1982) **R. A. GILBERT** (1995, 2003)

Diagrams are key components of a process documentation portfolio. The complexity of a modern production process has increased the process engineer's use and reliance on symbolic representations of a process and its unit operations. This section reviews diagram symbols and their role in binary logic diagrams. The complete description of a production process demands the use of several types of descriptive diagrams. The three important drawings that make up a specific process documentation package are the process function diagram (a site-specific binary logic diagram), the process flow sheet (a piping and instrument diagram [P&ID]), and a hardware diagram (a detailed equipment wiring diagram).

LOGIC DIAGRAMS

Consider a section of a chemical vapor deposition process that includes only a backup vacuum roughing pump. The process function diagram, Figure 1.12a, along with its accompanying process flow sheet, Figure 1.12b, are expected to provide all of the information required to understand the control scheme and general operation of this process subsystem. For the example illustrated, the process flow sheet indicates the equipment and instrumentation associated with the operation of the backup vacuum pump. However, a P&ID has its limitations. For example, this P&ID indicates that the vacuum pump is water-cooled but does not provide any plumbing details. In addition, the diagram makes little or no attempt to indicate the specific details of the reactor's backup vacuum control scheme. In fact, key control components, start and stop pushbuttons for example, are commonly omitted in P&IDs.

The role of the process function diagram and the binary logic diagram is to illustrate the overall control flow of the process. The diagram is a combination of logic and instrumentation symbols that reflects the overall control strategy. It also includes the details for the successful operation of the specified process equipment. For the example under consideration, the binary logic diagram indicates the operation requirements for the auxiliary vacuum pump. Note that the symbol for this pump doesn't even appear in the diagram; however, its operational characteristics with respect to the cooling water system are clearly indicated. In addition, the diagram does not provide for the reader or require the reader to possess knowledge of the complex and specialized circuit information associated with the assembly of the instrumentation presented. Its function is to provide an informed view

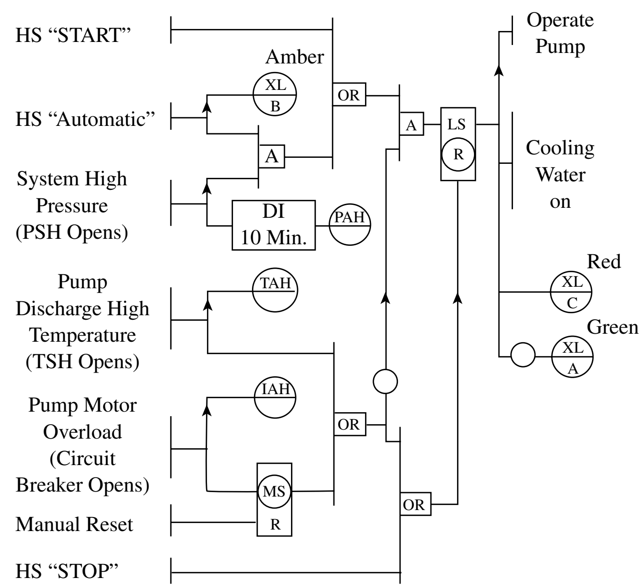


FIG. 1.12a
Logic diagram for standby vacuum pump with cooling water on and water valve (UV) open (see Table 1.12g).

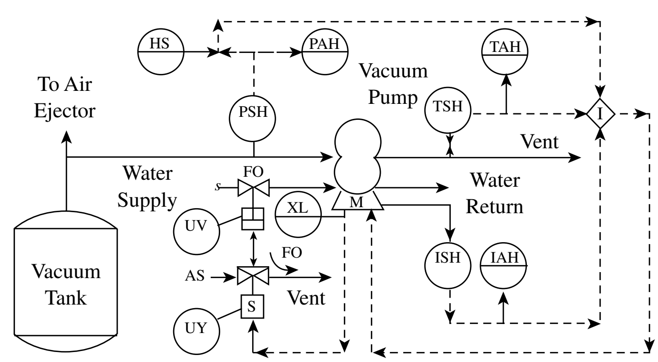


FIG. 1.12b
Control system for standby vacuum pump.

* Permission by the ISA to abstract from its standard, ANSI/ISA S5.2 Binary Logic Diagrams for Process Operators, is gratefully acknowledged.

of the process to operating personnel, maintenance workers, process engineers, and others who need to appreciate the expected behavior of the specified process equipment.

LOGIC SYMBOLS

Although the symbols selected to represent the logic operations associated with a control scheme are arbitrary, the choice of logic operations is universally consistent. All process function diagrams describe a process control scheme as a sequence and/or combination of the AND, the OR, the NOT, the NAND, and the NOR logic operations. Table 1.12c summarizes the ANSI/ISA standard symbols that describe these operations. The additional symbols in the table illustrate other commonly represented logic functions.

It is important to recognize that binary logic diagrams are "living" documents. Because they illustrate a process control procedure, binary logic diagrams also reflect any updates in process control strategy as well as the personalities of the process personnel who create and maintain the documents. As a result, it is likely that a selected (i.e., standard) set of diagram symbols will be modified within a company's various plant locations across the country, and completely different symbol sets may be adopted by different process industries.

Table 1.12d summarizes another common set of logic symbols. These symbols are often used in the description of digital electronic circuit operation. As a result, they are particularly popular with process technical personnel with computer science or electrical engineering backgrounds. Although this and other varieties of logic symbolism for binary logic diagrams exist, once the diagram reader's attention is focused on the logic operations represented, the significance and meaning of even unfamiliar symbols becomes apparent.

PREPARATION OF LOGIC DIAGRAMS

The number of logic operations involved dictates the arrangement of the symbols on a binary logic diagram. There are no specific rules as to symbol placement. Most process function diagrams depict the flow of information from top to bottom and left to right. Lines interconnect the logic symbols, and arrowheads might be added to lines where the input to output directions are not clear.

A summary of possible final control element operating states is often added to the diagram when the control action is not the result of one of two specific alternative states. For example, if the control action involves a control valve, it is not sufficient to indicate that the valve is not closed. This could mean that the valve is completely open or that the valve is in some position between almost closed and wide open. Since logic diagrams should be interpreted as literally as possible, in this example, the reader would accept that the diagram indicates that the valve was not closed but make no assumptions as to its degree of openness.

By contrast, if a logic diagram indicates that the result of the logic operations is the control of a constant-speed pump, it is sufficient to indicate the pump's state as either operating or not operating, i.e., the logic one or logic zero state. In this case, it would be the reader's responsibility to recognize that any variation in pump throttling was not the result of the control signal to the pump but rather due to the action of a separate signal to its accompanying valve. The two possible pump logic signals are responsible only for starting or stopping the transfer of energy to the pump. Any planned adjustment in the fluid flow rate is not the result of the logic one or logic zero binary signals delivered to the pump.

The binary logic diagram may have additional inputs to its logic symbols that do not represent actual process variables. For example, a process operation may be affected by loss of a component power supply. This would be especially true if the process equipment involved had electronic memory elements, and that memory was vital to the specific operation under consideration. Thus, the power failure might adversely affect the performance of one part of the process but not shut down its operation or directly influence the operation of other elements of the process. To take such situations into account, it would be necessary to consider the effect of that power loss on the logic of that component's operation and indicate what the new control action might be. In such cases, power or loss of power should be entered as a logic input to the diagram. The diagram should then indicate the expected sequence of events as if the power signal were just an additional binary process input to the entire process.

The indication of time delays and other time functions is common in binary logic diagrams. It is recommended for clarity that a single time function symbol be used to represent each single time function in the diagram. The use of one symbol to represent a sequence of unusual or complex time functions in immediate succession is often confusing to the reader. Even the correct use of NOT operations combined with a time function is confusing and often leads to the diagram reader reaching the opposite conclusion intended from these symbol combinations.

It is quite common for the binary logic diagram for a process to require more than one page of drawings. When the diagram has more than one page, each sheet should be clearly labeled with the plant location, the process identification, the last update to the diagram, and which page out of the total this sheet happens to be. For example, each sheet of an 11-page logic diagram might be labeled:

"Cheese Curd Vacuum Packing Process"
Tug Hill Packing Plant

The fifth sheet for this process diagram would have the additional information:

"Auxiliary Vacuum Pump Control"
Last Update: March 15, 2003
Sheet 5 of 11

TABLE 1.12c
Binary Logic Diagram Symbols

<table>
<tr><th>Function</th><th>Symbol</th><th>Definition and Truth Table</th><th>Example</th></tr>
<tr><td>INPUT</td><td>(Input statement) ├─

Statement may be preceded by instrument balloon with tag number.</td><td>Input for logic value or sequence of values.</td><td>Start chemical injection by pushbutton.

Start injection manually ├─</td></tr>
<tr><td>OUTPUT</td><td>(Output statement) ├─

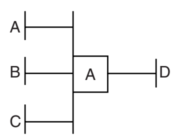
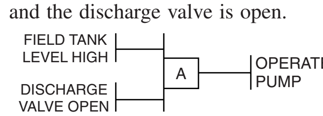
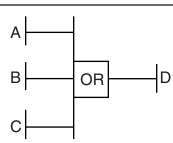
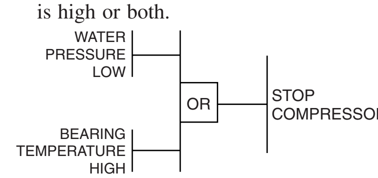
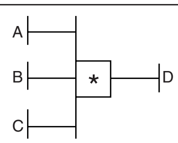
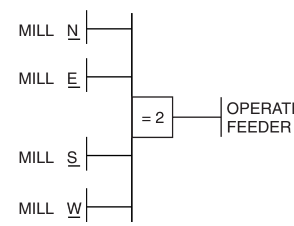
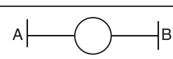
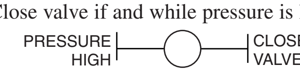
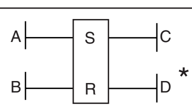
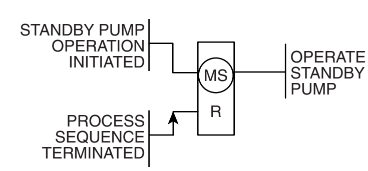
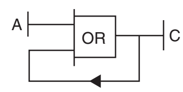
Statement may be followed by instrument balloon with tag number.</td><td>Logic value provided as an output signal because of an input logic sequence.</td><td>The logic sequence causes drawoff to cease.
Stop drawoff ├─</td></tr>
<tr><td>AND</td><td></td><td>Output D is active only when inputs A, B, and C are all active.
<table>
<tr><th></th><th></th><th colspan="2">C</th></tr>
<tr><th>A</th><th>B</th><th>O</th><th>I</th></tr>
<tr><td>O</td><td>O</td><td>O</td><td>O</td></tr>
<tr><td>O</td><td>I</td><td>O</td><td>O</td></tr>
<tr><td>I</td><td>O</td><td>O</td><td>O</td></tr>
<tr><td>I</td><td>I</td><td>O</td><td>D</td></tr>
</table></td><td>Operate pump if feed tank level is high and the discharge valve is open.
</td></tr>
<tr><td>OR</td><td></td><td>Output D is active only if and while one or more of inputs A, B, and C are active.
<table>
<tr><th></th><th></th><th colspan="2">C</th></tr>
<tr><th>A</th><th>B</th><th>O</th><th>I</th></tr>
<tr><td>O</td><td>O</td><td>O</td><td>D</td></tr>
<tr><td>O</td><td>I</td><td>D</td><td>D</td></tr>
<tr><td>I</td><td>O</td><td>D</td><td>D</td></tr>
<tr><td>I</td><td>I</td><td>D</td><td>D</td></tr>
</table></td><td>Stop compressor if cooling water pressure is low or bearing temperature is high or both.
</td></tr>
<tr><td>QUALIFIED OR</td><td>
*Insert numerical quantity (see "Definition").</td><td>Output D is active only if and while a specified number of inputs A, B, and C are active.
Mathematical symbols shall be used, as appropriate, in specifying the number, e.g., ≥3, to denote 3 or more.</td><td>Operate feeder if and while two and only two mills are in service.
</td></tr>
<tr><td>NOT</td><td></td><td>Output B is active only if and while input A is passive.</td><td>Close valve if and while pressure is low.
</td></tr>
<tr><td>FLIP-FLOP MEMORY</td><td>
*If output D is not used, it shall not be shown.</td><td>S denotes set memory and R denotes reset memory.
Output C is active as soon as input A is active. C remains active, regardless of the subsequent state of A, until the memory is reset, i.e., terminated when B is active. C remains passive, regardless of the subsequent state of B, until A causes the memory to be set.
Output D, if used, is active when output C is passive and D is passive when C is active (i.e., a signal flip-flop).
Input-Override Option If inputs A and B are active simultaneously, and if A is then required to override B, then S should be encircled, i.e., if B is to override A, then R should be encircled.
Loss-of-Power-Supply Option Required action of the memory on loss of power shall be symbolized by modifying the set letter, S, as follows:</td><td>If standby pump operation is initiated, the pump shall operate, even on loss of the logic power supply, until the process sequence is terminated. The pump shall operate if START and STOP commands exist simultaneously.
</td></tr>
</table>

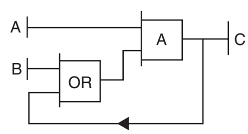
TABLE 1.12c Continued
Binary Logic Diagram Symbols

<table>
<tr><th>Function</th><th>Symbol</th><th>Definition and Truth Table</th><th>Example</th></tr>
<tr><td></td><td></td><td>

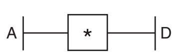
Modified Symbol	*Required Memory Action on Loss of Power*
S (unchanged)	Was not considered by logic designer
LS	Memory lost
MS	Memory maintained
NS	Is not significant, no preference

The use of a logic feedback to symbolize a memory that is lost in the event of loss of power although depicted below, is not recommended. Thus, the following shall not be used:





</td><td></td></tr>
<tr><td>TIME ELEMENT</td><td>



*Insert symbolism for specific functions and time interval (see “Definitions”).

</td><td>

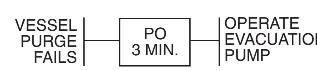
Basic Method

This uses the following specific symbols:

Symbol	Meaning
DI	Delay Initiation of output. The continuous existence of input *A* for a specified time causes output *B* to become active when the time expires. *B* becomes passive when *A* becomes passive.
DT	Delay Termination of output. When input *A* is active, output *B* becomes active. When *A* becomes passive for the specified time, Output *B* will become passive (terminate) as well.
PO	Pulse Output. If Input *A* becomes active and then immediately passive, output *B* is active for specified time.

</td><td>

If a Pressure Alarm High is active (in alarm) for 4 sec, the auxiliary vacuum pump is engaged. The pump shuts down when the PAH becomes passive.



When the auxiliary vacuum pump is active (running), the water jacket pump is also engaged. When the auxiliary vacuum pump is shut down, the water jacket pump remains on for 5 min before it shuts down. If vacuum pump is started before 5 min is up, water jacket pump stays on.

If vessel purge fails even momentarily, operate evacuation pump for 3 min and then stop the pump.

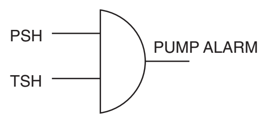
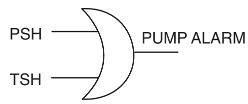
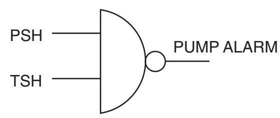
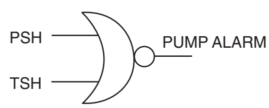
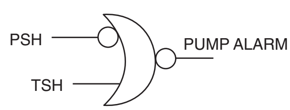
</td></tr>
</table>

In addition to this detailed labeling of each sheet of the diagram, care must be taken to identify lines that come from or go to other sheets of the drawing. There are no standards for accomplishing this, but one good practice is to repeat the label on a line that leaves one sheet as that line begins again on another sheet. Another popular approach to the task is to identify each sheet entrance and exit line with its source and destination. Numbering all exit lines from one sheet from top to bottom and then identifying each line that enters a new sheet in the same manner is the usual approach. With this numbering system in hand, each line entering or leaving a sheet can be identified with its source and destination address.

For example, suppose a diagram line that leaves sheet 5 of a binary logic diagram was labeled 5–7/8–3. This label would identify that logic signal as exit signal 7 from sheet 5 and indicate that it was going to enter sheet 8 as the third external input from the top of the page. To complete the redundancy for this numbering system, 8–3/5–7 would be the label on the third external input line from the top of sheet 8. This tag would confirm that the signal in question indeed was the third external sheet source input signal for sheet 8 and that this particular signal was also the seventh off-the-sheet output from sheet 5. Letters and numbers are often interlaced in this identification scheme, but the source, destination, and redundancy aspects of the system are always maintained.

In summary, the creation of a binary logic diagram requires specific knowledge of how the process is controlled. In addition, this type of process function diagram must be clearly

TABLE 1.12d
An Alternate Set of Logic Symbols

Function	*Symbol*	*Definition*
AND		If PSH and TSH are both at logic 1, then output to pump alarm is at logic 1
OR		If either PSH or TSH is at logic 1, then output to pump alarm is at logic 1
AND		If PSH and TSH are both at logic 1, then output to pump alarm is at logic 0
OR		If either PSH or TSH is at logic 1, then output to pump alarm is at logic 0
Mixed		If PSH is at logic 0 or TSH is at logic 1, then output to pump alarm is at logic 0

Note: When logic 1 is assigned to the high physical signal value, the output of an AND device is active when it is at logic 1.

TABLE 1.12e
Intermediate Position Labels

Device State	*Definition*
Open position	Device 100% open
Not open position	Device less than 100% open (may or may not be closed)
Not closed position	Device more than 0% open
Closed position	Device that is 0% open
Intermediate position	Device at specific position (greater than 0% but less than 100% open)
Not-at-intermediate position	Device at position above or below the specified intermediate position (0% or 100% open)

labeled and religiously updated. Below are several common-sense guidelines that will help develop a good diagram:

1. Develop the logic diagram in such a way that it presents results in a positive logic format. For example, if the result of the logic operation is to completely open a valve, it should be so stated and not be stated as being "not closed."
2. Use a specific separate label to indicate conditions of final control elements that have more than two responses to binary signals. Table 1.12e defines labels for devices that have open, closed, and intermediate positions.
3. Label each input and output of each set of logic operations on every sheet of the diagram.
4. Label each external input or output for each sheet of the diagram so that its source and destination are clearly understood.
5. Adopt a standard set of logic symbols and instrument labels (see Section 1.1), and resist the temptation to modify your logic symbols for the "special situation" that inevitably develops. If the situation cannot be initially explained with relatively simple combinations of AND, OR, NOT, NAND, and NOR symbols, think about it some more until it can be done that way.
6. Avoid using ladder diagrams or ladder diagram logic symbolism. These types of icons are useful when installing or troubleshooting the control scheme but are only routinely used and understood by a small percentage of the technical personnel who need to know about the process.

These useful rules of thumb all follow from one "golden rule."

If the binary logic diagram does not clearly, easily, and unambiguously indicate what is going on in the process, then the diagram is of marginal value and should be reworked until it does.

Following this rule will increase the chances of having the diagram match what it is intended to say with what it actually says.

DIAGRAM INTERPRETATION

Even if authors follow the "golden rule" when creating logic diagrams, interpreting logic diagrams is not always an easy assignment. Consider the example process presented in this section. Its process function diagram, Figure 1.12a, describes the operation of a standby vacuum pump system for a process that needs to maintain a specified pressure in the vacuum tank illustrated in Figure 1.12b. Information about the panel instrumentation indicated in Figure 1.12b is provided in Table 1.12f. Complete details about the other instrumentation symbols used in Figure 1.12b are available in Section 1.1.

The logic diagram shown in Figure 1.12a indicates that the backup pump control is based on the binary value from seven independent process input signals. By contrast, a close inspection of the diagram reveals the existence of only one independent output signal. This example's high ratio of input to output signals is typical of most binary logic diagrams, indicates the reason why logic diagrams might be difficult to interpret, and suggests that a good way to analyze the entire diagram is to study it from right to left.

TABLE 1.12f
Control Panel Instrumentation

Panel Symbol	Instrument Function
HS	Manual control switch for pump operation. Three momentary-contacts for Start, Automatic, and Stop actions.
PAH	Alarm if vacuum fails. Alarm sounds 10 min after backup vacuum pump start request.
TAH	Temperature alarm if pump discharge temperature rises above normal operating value.
XL-A	Green pilot lamp when bright indicates pump motor circuit breaker is not closed, i.e., pump cannot operate.
XL-B	Amber pilot lamp when bright indicates pump is ready for automatic start.
XL-C	Red pilot lamp when bright indicates pump motor circuit breaker is closed, i.e., the pump can operate.

Note: XL-C is not connected to a pump motor feedback signal. It cannot confirm that motor is on.

As stated above, the condition of all four output logic signals in Figure 1.12a (i.e., the red and green pilot lamps, the cooling water control value, and the pump operation signal) depends on the logic state of a single output. This is represented as the logic output from the "memory lost on loss of power" memory operation of the LS flip-flop located on the upper far right side of the diagram. The logic one signal on this output will cause the pump to be engaged, the cold-water control valve to open, and the red closed motor circuit-breaker light to glow. The logic signal inversion represented by the circle symbol between the flip-flop output and the green not-closed motor circuit-breaker light (XL/A) assures that the green light is not glowing when the flip-flop output is at logic one.

It should be clear at this point in the analysis of Figure 1.12a that the input signals to the LS flip-flop are the key to understanding the operation of the backup vacuum pump. The diagram indicates that this flip-flop is controlled by two input logic signals: a signal that comes from an AND operation (the "pump run" AND) and a signal that comes from an OR operation (the "pump reset" OR). It is also clear, but not necessarily appreciated by the reader, which of these two input signals is more important.

The output of the "pump reset" OR device goes to the reset terminal of the memory lost on loss of power to the flip-flop. If this OR output is at logic one, the flip-flop's output resets to logic zero. This logic signal forces the pump to shut down, the cold water control valve to close, and the red panel indicator (XL/C) to go off.

The circle around the letter R in the memory lost on loss of power flip-flop symbol indicates that a logic one output from the "pump reset" OR will initiate the flip-flop reset action even if the output of the "pump run" AND device is at logic one. Since the HS "STOP" signal is directly connected to one of the inputs of the "pump reset" OR, placing the hand switch in the HS "STOP" position allows the pump to receive the logic zero reset signal from the LS flip-flop and stop running.

Although not directly connected to the "pump reset" OR, activation of either the PUMP MOTOR OVERLOAD CIRCUIT BREAKER or the PUMP DISCHARGE HIGH TEMPERATURE (TSH) alarm produces logic signals that generate the same control action as the HS "STOP" input. These two control signals feed to an OR that is connected to the other input of the "pump reset" OR. Therefore, a logic one output signal from this OR produces the same response in the "pump reset" OR as a logic one signal from the HS "STOP".

The "memory maintain on loss of power" flip-flop is located in the lower left portion of Figure 1.12a. It is arranged so that, if both its inputs are activated simultaneously, the PUMP MOTOR OVERLOAD CIRCUIT BREAKER signal overrides the MANUAL RESET signal. In other words, the pump cannot start again until the pump motor overload has been cleared. Once that pump motor circuit problem no longer exists, a logic one MANUAL RESET signal will produce a logic zero on the memory-maintain flip-flop. If the TSH sensor is not in alarm (i.e., at logic zero) or the HS "STOP" is not engaged (i.e., at logic zero), then the logic zero output from the flip-flop is passed directly to the reset input of the memory lost on loss of power flip-flop. Although, at this point, the flip-flop's output still remains at logic zero, and the pump remains off, the pump will engage the next time the "pump run" conditions are satisfied.

"Pump run" operating scenarios are determined by the output of an AND operation. This "pump run" AND is the other input connected to the "memory lost on loss of power" flip-flop. If the output of the "pump run" AND is at logic one, the pump will be turned on. The only constraint on this logic

TABLE 1.12g
Control Valve Operating Sequence

Current Vacuum Pump Status	Motor Circuit Breaker Auxiliary Contacts	Solenoid Valve (UY) Coil	Control Valve (UV)		
			Actuator	Port	Water
Off	Closed	Energized	Pressurized	Closed	Off
On	Open	De-energized	Vented	Open	On

one "pump run" signal is if the reset input to the memory-loss flip-flop is already at logic one. Under that condition, the "pump run" signal will not override the reset, and the pump will not start.

There are two inputs to the "pump run" AND operation. The HS "START," HS "AUTOMATIC," and SYSTEM HIGH PRESSURE (PSH) process signals identified in the upper left corner of Figure 1.12a all eventually feed their logic signals to one of these inputs. Basically, the pump will engage if the hand switch is in the HS "START" position or if it is in the HS "AUTOMATIC" position and the PSH is in alarm, i.e., at logic one. An operator panel alarm function is also provided when PSH goes into alarm. If the pressure alert is not corrected within 10 min (i.e., the backup vacuum pump has not returned the vacuum tank to an acceptable vacuum condition), a panel alarm, PAH, will be activated.

The other input to the "pump run" AND is a gate signal from the TSH and PUMP MOTOR OVERLOAD CIRCUIT BREAKER logic section of the diagram. This gate signal is at logic one only when the pump is not in a temperature or overload circuit alarm condition. If either such alarm condition exists, the gate input to the "pump run" AND would be at logic zero and the pump would not run even if the reset input to the "memory loss on lost power" flip-flop were at logic zero.

ACTIVE AND PASSIVE LOGIC

The example process provided in this section is adequate to illustrate the important aspects of drawing and reading a process diagram. This specific logic diagram is typical with respect to elements in the diagram but not in the number of these elements. To read diagrams that have many more components and interactions among them, it is convenient to initially identify the input and out logic signals as active or passive and then further classify the logic device inputs and outputs as active high or active low. For the example presented in Figure 1.12a, the vacuum pump is active when a logic one signal is presented to that pump. By contrast, the XLA indicator is active when the logic zero signal is presented to it. Thus, the vacuum pump input terminal is identified as an active high input while the XLA indicator input terminal is an active low input. When the vacuum pump is not running, the signal to the pump is the passive logic signal; when the XLA indicator is on, the signal to that device is the active signal. Thus, the entire logic diagram can be initially reviewed by first establishing what the active and passive signals are for each device represented in the diagram and then determining if the input and output terminals for each device are identified as active high or active low terminals.

For this standby vacuum pump example, the AND device that accepts signals directly from the HS "automatic" and PSH field elements has two active high inputs and one active high output. The AND device that controls the LS flip-flop has an active high input, an active low input, and an active high output. The three OR devices each have two active high inputs and one active high output. The efficiency of this identification scheme is appreciated once it is realized that the output of an AND logic device is active only when all its inputs are active, while the output of an OR device is active when any of its inputs are active. Thus, it becomes clear very quickly that the OR device in the lower part of the diagram that deals directly with high-temperature and motor-overload situations can prevent the HS "automatic" input from activating the standby vacuum pump. It also becomes clear that, when the output of that OR device is active, the LS flip-flop is reset, and the vacuum pump would stop if running. In other words, the vacuum pump becomes passive.

FINAL CAUTION

Now that the function diagram in Figure 1.12a has been reviewed, a few concluding remarks are in order. A complete understanding of a process requires all three types of diagrams in the process documentation package. For the example sited, no attention was focused on the hardware diagram. In fact, assumptions were made to facilitate the explanation of Figure 1.12a. For example, Figure 1.12a does not indicate how the green not-closed motor circuit-breaker light (XL/A) is wired. If it uses a pull-up resistor type wiring scheme, then the light is off when the motor is running. However, if XL/A is wired directly to its source, then the inverted logic one from the memory lost on loss of power LS flip-flop will allow the XL/A light to glow. If this were the situation, many binary logic diagram authors would draw a line over the XL/A symbol to indicate that the green not-closed motor circuit-breaker light had an active low input. This would indicate that the light would glow when it detected the low logic signal. Always check the wiring diagram to determine the active/passive signals for all sensors and final control elements included in a binary logic diagram.

The biggest danger when reading a binary logic diagram is projecting what "should" be there instead of reading the diagram as it is written. There are often different views as to how to implement the specific details for a process control strategy. The example process discussed in this section is no exception to that observation. Constructive criticism about a control scheme is an important aspect of improving the control of a process. However, it is important to distinguish between not liking the logic presented in a logic diagram and mentally refusing to read the logic as it is presented in the diagram. The responsibility of binary logic diagram creators is to produce unambiguous diagrams that explain the process's current control logic. The responsibility of binary logic diagram readers is to interpret the diagram without altering its mental image with personal bias about the control scheme described. Once the creator and reader of a logic diagram agree as to what the diagram says, then it is time to discuss the merits of the current control scheme and consider any reader-suggested alterations.

Bibliography

ANSI/ISA S5.2, 1976 (reaffirmed July 13, 1992), Binary Logic Diagrams for Process Operations, ISA, Research Triangle Park, NC.

Platt, G. and Barrikka, N.E., *Process Control: A Primer for the Nonspecialist and the Newcomer,* 2nd ed., ISA, Research Triangle Park, NC, 1998.

Course No. 205, Fundamentals of Process Instrumentation, Job Training Systems, Inc., Unionville, PA.

ISA Dictionary of Measurement and Control, 3rd ed., ISA, Research Triangle Park, NC, 1995.

Measurement and Control Basics, 3rd ed., ISA, Research Triangle Park, NC, 2002.

Programmable Controllers, 3rd ed., ISA, Research Triangle Park, NC, 2000.