The purpose of this SAMPLE document is to show in the public domain the SIF General Design Background that should be considered as general requirements for project development.

LIUTAIO
"Functional Safety Services"

The information in this document should be part of the PROJECT "Safeguarding Philosophy" or "Safeguarding Specification". And it should be included as a mandatory requirement for developing of:
Safety Requirements Specification (SRS), SIL verification, SIS-DCS integration, Maintenance Override Switches (MOS), Proof Test, etc.

Table of Contents

1. Document purpose

To take care of SAFETY, any industrial project shall develop the "Hazard and Operation Studies" (HAZOP) and to stablish which are the Actions/Modifications/Adjustments in the plant design to reduce the operation risk up to the acceptable limits.
The Actions/Modifications/Adjustments may include design changes, operation procedure changes, and "Safety Instrumented Functions" (SIFs).

The purpose of this document is to stablish the knowledge base line to describe a "Safety Instrumented Function" (SIF), SIF components, and required minimum SIS-DCS integration for SIF proper implementation.

ALL described principles in this document can be fully applicable to electrical/electronic/programmable technology, nevertheless the same principles can be applied to other technologies, like: Hydraulic, Pneumatic or Hybrid.

Information like the one described in this document should be included in the project documents like "Safeguarding Philosophy" or "Safeguarding Specification".

2. General "Safety Instrumented System" (SIS) description

During normal operation, a Machine, Equipment or Process Plant is in the **NORMAL state**. It means, it is in normal production, and no safety issues are in progress.

The "Safety Instrumented System" (SIS) is a protection system used to implement one or more "Safety Instrumented Functions" (SIFs), which will set a plant or equipment to its **SAFE state**, prior to any hazard happens. A SIS is composed of any combination of sensor(s), other devices, logic solver(s), and final safety elements(s).

3. SIS "Human Machine Interface" (HMI)

The "Human Machine Interface" (HMI) is the panel, computer screen, or display that is used by a Console/Field Operator to visually monitor and command the operation of a Machine, Equipment or Process Plant.

For small plant installations, VENDORs use to provide a panel with graphic interface as HMI. However, the practice indicates that the HMI and SIS capabilities are separate equipment.

For bigger plant installations, a bigger HMI is required. It is a plant owner decision to keep separate the Operation and Safety HMI, or to integrate them.

Regardless the plant size, normally a separate safety panel is provided to allow Console/Field Operator to manually initiate the most important safety actions through commanding of physical push buttons and switches. Nevertheless, nowadays is possible to incorporate many of the safety actions through soft buttons/switches in the HMI, in order to reduce the safety panel size.

4. General "Safety Instrumented Function" (SIF) description

The purpose of a "Safety Instrumented Function" (SIF) is to avoid a machine, equipment or process plant to be operated under conditions that will result in a harm of the personal safety, the environment or the machine/equipment/process plant itself.

From the instrumentation point of view, a SIF defines which the "SIF Input Signal(s)" is (are) will be used to trip an action(s) (or Command(s)) to be sent through the "SIF Output Signal(s)" to set the "Final Safety Element(s)" in the SAFE state; in order to move the related plant, machine or equipment to a SAFE state as well.

The SIF typical elements are: Sensors and Transmitters (which generate the SIF input signals), Trip Criterion, Voting Logic, Decision logic, Logic Solver (which generates the SIF output signals), and Trip device plus Final Safety Element.

Some other SIF elements are not listed in this section for making explanation simple, but they must be considered in the SIF design. These other elements are, but they are not limited to: input card, output card, interposing relay, barriers, "Uninterrupted Power Supply" (UPS), "Hydraulic Power Source", etc.

Review following figures for illustration:

Figure 1 describes the most simple SIF structure with only one(1) input and one(1) output.

Figure 2 describes the simple SIF with voting logic with several inputs and voting logic among inputs.

Figure 3 describes the SIF structure with multiple inputs and multiple Voting Logics.

Figure 1 – Simple SIF typical structure with only one(1) input and only one(1) output

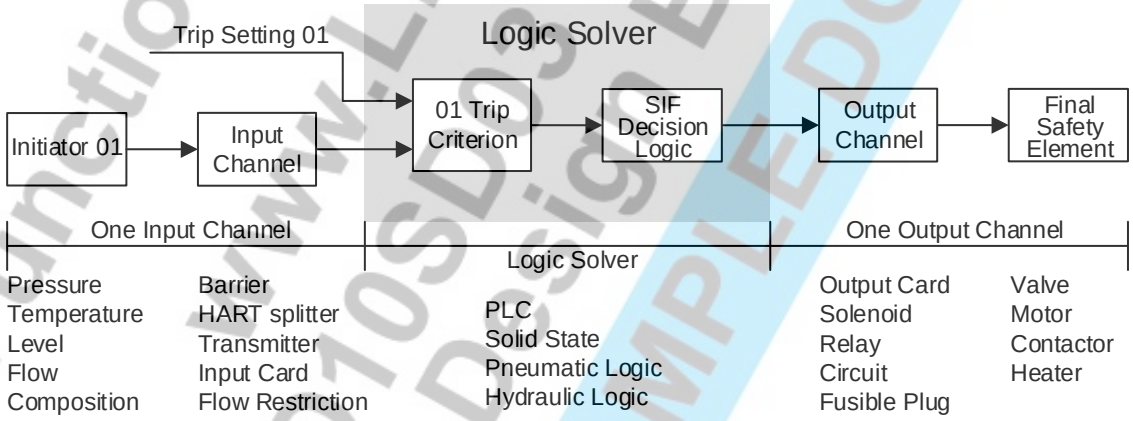


Figure 2 – Simple SIF structure with voting logic with several inputs and voting logic among inputs

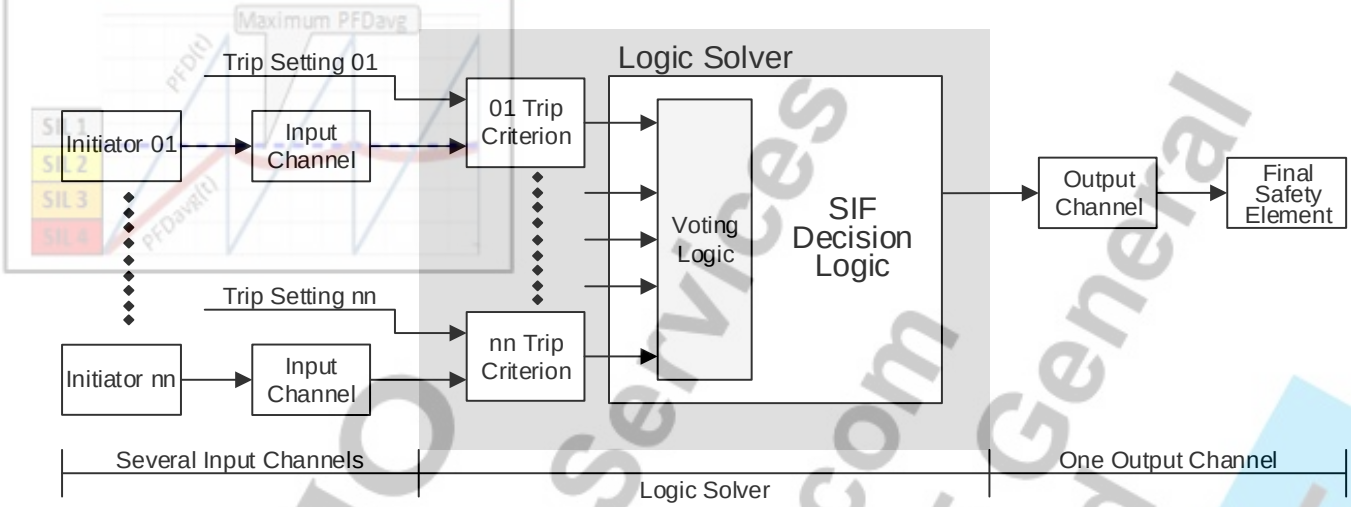
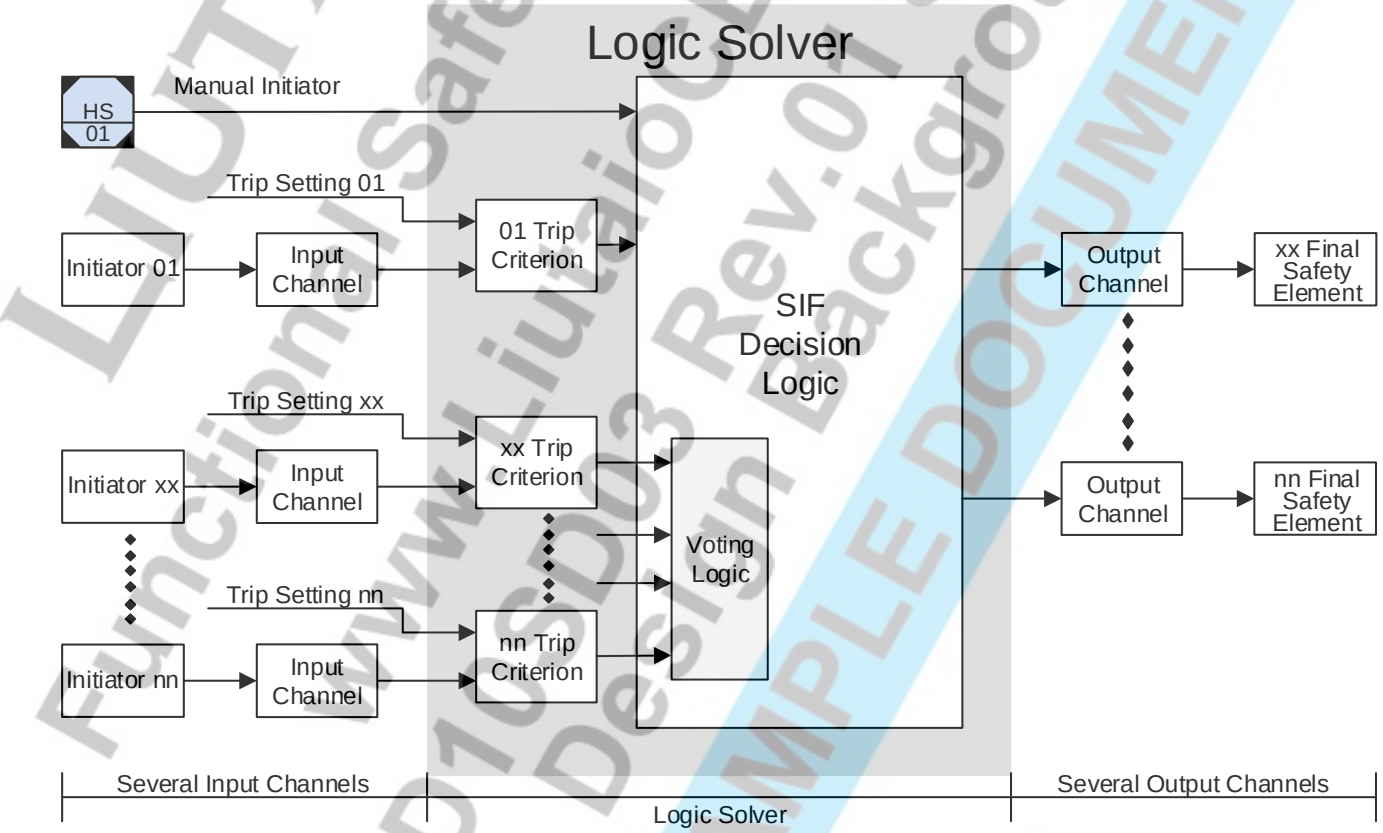


Figure 3 - SIF structure with multiple inputs and multiple Voting Logics



The "**Initiator**" is the monitored process variable that will determine if the SIF will trip or not.
The "Initiator" signal also can come from a manual push button.

The "**Input Channel**" is monitoring the process variable continuously and transmitting the process state to the "Logic Solver" for determining when the safety actions shall be executed.

An "Input Channel" can handle signals of Pulse, Analogue or Digital type.

Different technologies and devices of different types can be used to implement an "Input Channel", for example:

Sensor — is monitoring the process, fetching a process variable value continuously. There are many different technologies to measure a process variable, and also for measuring each process variable type.

Transmitter — reads the process variable value from the Sensor, next amplify/normalize/convert the process variable value to a standard communication protocol and sends that value (or SIF Input Signal) to the "Trip criterion". Nowadays some communication protocols also send "Sensor/Transmitter" configuration and statuses information. Sometimes the "Sensor" and the "Transmitter" are contained in the same device. Example of this is a nowadays pressure transmitter.

HART splitter (or Multiplexer) — It is a signal splitter that converts one DC current or millivolt input into two(2) isolated proportional 4-20 mA control signals. Power can be received from one or both output loop current.

Input Barrier — A barrier (Intrinsically Safe, or not) is used to provide protection to a device mounted in a hazardous location. The basic Barrier objective is to limit the current passing through the connected safety elopements in the hazardous area, in order to avoid ignition sources. To limit the current, the barrier can be based on multiple series resistors (assuming that resistors always fail open), multiple Zener diodes to re-route the current to earth, or opto-couplers.

The "**Trip criterion**" is a logic to apply to the information that is coming from one(1) "Initiator" to determine if such process variable is in the NORMAL or in the TRIP state.

The simpler "TRIP criterion" consists in comparing an analogue "Initiator" value against its "Trip value" or "Trip setting". The comparison just verifies if the process value is greater or lesser than the "Trip setting".

If the SIF has just one[1] "Trip criterion", the "Trip criterion" output is connected directly to the "SIF Decision Logic" (see Figure 1).

If the SIF has more than one[1] "Trip criterion", such "Trip criteria" shall be connected to other "Trip criterion" or to a "Voting Logic", before passing the criteria result to the "SIF Decision Logic" (see , Figure 2 and Figure 3).

The "Trip Criterion" can be implemented:

- Inside the "Logic Solver" as a configured logic, or
- Outside the "Logic Solver" inside or part of a Device.

The "**Voting logic**" is a logic to be applied to several "Trip criteria" outputs, in order to determine if their condition is in the NORMAL or in the SAFE state.

Typical "Voting logics" are: 1oo1, 1oo2, 2oo2, 2oo3, 1oo1D, 1oo2D, 2oo3D, etc.

"Voting logics" with diagnostic (like: 1oo2D, 2oo3D, etc.) are able to handle failure statuses from the "Initiator" and/or "Input Channel" , and transmit these statuses to the "Logic Solver", in order to use this information to:

a) Notify Operator that a problem appeared in the SIF, and MAINTENANCE MUST be applied, and
b) if such failure may lead to make the SIF to fails on demand, then the SIF implementation shall trip after the "Mean Time To Restoration" (MTTR).

One[1] or more "Input Channels", "Trip criteria" and "Voting logics", which are in the NORMAL or SAFE state, can be connected to the "**SIF Decision Logic**". This logic determines to set in the NORMAL or SAFE state the SIF's "Final Safety Element(s)".

The "**Logic Solver**" is a device, part of a SIS, that can execute the "Safety Logic" (many "TRIP criteria", "Voting Logics" and "SIF Decision Logics"). Nowadays the "Logic Solver" could be part of a module in the DCS (PCS), or an independent module based on electrical, electronic, mechanical, pneumatic or hydraulic technology. Sometimes a hybrid "Logic Solver" can be used.

The "**Output Channel**" is transmitting the signal from the "Logic Solver" up to the "Final Safety Element", in order to make it, or allow it to move from the NORMAL to the SAFE state when the "Logic Solver" indicates to do it.

According to the nature of the "Output Channel" that is connected to the "Trip Device", its function in the SAFE state is one of the following:

a) To interrupt the mechanism that makes the "Final Safety Element" to remain in its NORMAL state.
b) To start a mechanism that moves the "Final Safety Element" from its NORMAL state up to its SAFE state.

Different technologies and "Trip Devices" of different types can be used to implement an "Output Channel", for example:

Output Barrier	See "Input Barrier" above.
Solenoid valve (SOV)	It is an electromechanically operated valve, is controlled by an electric current through a solenoid. In the case of a two-port valve the flow is switched on or off; in the case of a three-port valve, the outflow is switched between the two outlet ports.
Interposed Relay	It is means a relay that is used to separate or put a barrier between two circuits. It normally consists of a coil that is Energized (or De-Energized) by either AC or DC power and activates contacts that are used to trigger another circuit.

The "**Final Safety Element**" (FSE) is normally a valve (+actuator) that has to be Opened/Closed/Opened, a motor that has to be set in the On/Off state, etc.

During normal operation, the FSE is in the NORMAL state. To protect the related Machine, Equipment or Process plant against to an identified Hazard, the "Logic Solver" sets the FSE to the SAFE state.

Different technologies and devices of different types can be used as a FSE, for example:

Block valve	Or "Shutdown Valve", is a mechanical device that is used to commonly close or block a flow stream.
Blowdown valve	It is a mechanical device that is used commonly to open a flow path in a stream.
Contactor	A "Contactor" is an electrically controlled switch (relay) used to provide or to cut electrical power supply to the electrical motor of a pump, compressor, blower, electrical heater, etc. Unlike a circuit breaker, a "Contactor" is not intended to interrupt a short circuit current.

4.1 Trip Pre-Alarm or Alarm

An Alarm is a message that requires the Console Operator attention and subsequence actions.

An ALARM can be implemented ONLY if by design enough time is foreseen to allow the Console Operator to react and manually initiate a corrective action in DCS or SIS. **ELSE NO ALARM IS REQUIRED**, because it is useless in practice.

A "Trip Pre-Alarm" follows the same requirements of an ALARM.

An ALARM can be implemented in DCS, in SIS, or both. Project directive or philosophy shall indicate which is the preferred location for ALARMs' implementation.

There is a difference between ALARM implementation and ALARM log:

- ALARM implementation refers to the location (DCS, SIS, other) where the logic to identify the ALARM is locate.
- ALARM log refers to the location where the alarm is notified and recorded. Normally DCS.

5. Additional SIF components for industrial applications

Following good engineering practices, and to satisfy the nowadays safety standards (IEC 61508, ISA 84.00.02), the implementation of a "Safety Instrumented Function" (SIF) shall include additional components for proper integration with Control and Operation, and for properly testing the SIF devices. These Additional SIF components shall be implemented into the "Logic Solver", as part of DCS (Logic and/or HMI), and/or as part of design & physical installation of Initiators, Final Safety Elenent(s), Input/Output Channels.

For describing the Additional SIF components, a simplification of Figure 1 is used as starting point. See Figure 4.

The Additional SIF components are listed following: (see Figure 5)

Additional components per SIF:

- Manual Initiator
- Manual Input MOS
- Start-up Bypass
- Reset Function
- Interlock Logic

Additional SIF components per Input:

- Input Proof Test Logic
- Input MOS

Additional SIF components per Output:

- Output Proof Test logic
- Output MOS

Figure 4 – Starting point for describing Additional SIF components

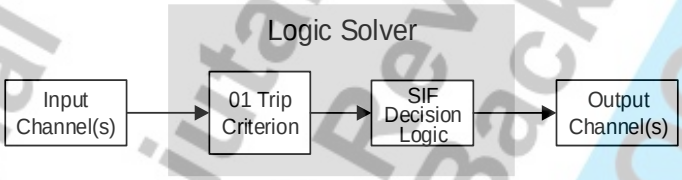
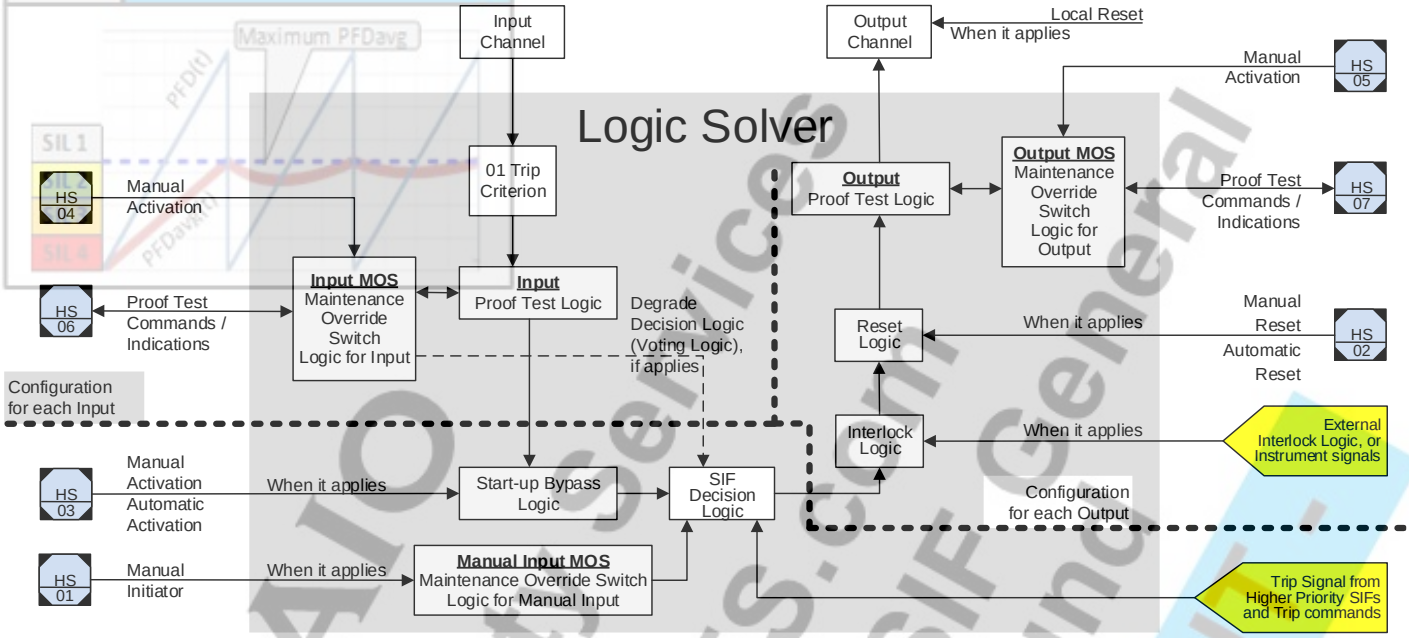


Figure 5 - Additional SIF components



NOTE: in Figure 5, design and implementation of each "Input Channel" and "Output Channel" **MUST BE UNIQUE**. This means that reset logic, proof test, MOS, etc. that are related to an Input/Output MUST BE configured/implemented only once. **NO DUPLICATION** of functionalities SHALL BE allowed per "Input Channel" and "Output Channel".

5.1 Additional components per SIF

5.1.1 Start-up Bypass

Sometimes to startup a Machine, Equipment or Process plant, it shall be possible to operate in such conditions that in normal operation will initiate a TRIP. For example:

Example 1: A gas separator operation MUST BE within a pressure range, else a hazard can occur, and a SIF shall close a safety valve in the separator inlet to avoid such hazard. But, to startup the gas separator initially the pressure is below the operation range, so in SAFE state and the SIF is keeping the safety valve in the SAFE state (Closed).

So, in order to allow the gas separator startup, a "Start-up Bypass Logic" shall be Activated, to make the SIF to open the safety valve during startup to pressurize the gas separator.

Once the pressure is within the operation range the "Start-up Bypass Logic" is Deactivated, and since the pressure is already in NORMAL state the SIF keeps the safety valve in the NORMAL state (Opened).

5.1.1.1 Implementation

To implement a "Start-up Bypass Logic" as part of a SIF is required when the override the SIF to set the FSE in the NORMAL state for a Machine, Equipment or process plant startup purpose.

After the Machine, Equipment or process plant is in operation and the SIF "Initiator" value passes the "Trip Setting", then the "Start-up Bypass Logic" shall be Deactivated and SIF starts to perform.

The "Start-up Bypass Logic" can be commanded manually or automatically.

Manually means that the Console Operator MUST Activate/Deactivate the logic manually through the DCS-SIS HMI.

Automatically means:

a) When SIS declares that the Machine, Equipment or process plant IS NOT in operation, then the "Start-up Bypass Logic" shall be Deactivated automatically.

b) During the Machine, Equipment or process plant startup, when the SIF "Initiator" value passes the "Trip Setting", then the "Start-up Bypass Logic" shall be Deactivated automatically, and the SIF starts to perform.

Regardless if the "Start-up Bypass Logic" is commanded manually or automatically, if a command tries to Activate the "Start-up Bypass Logic" while the Machine, Equipment or process plant startup, such command **MUST BE** rejected by the "Start-up Bypass Logic".

The "Start-up Bypass Logic" can be activated again ONLY after SIS declares that the Machine, Equipment or process plant IS NOT in operation.

A "Start-up Bypass Logic" is commanded by Soft-buttons or by Hard-buttons. The minimum requirements SHALL BE:

- For Hard-Buttons, to use a two-position Switch with Light SPDT. One position to Activate the bypass, and the other one to Deactivate it. The light hall indicate if the bypass is in fact activated or not.

- For Soft-Buttons, HMI push button with confirmation message.

NOTE: The Machine, Equipment or process plant startup logic or procedure shall include to Activate the SIF's "Start-up Bypass Logic" as a startup PERMISSIVE.

5.1.2 Reset Function

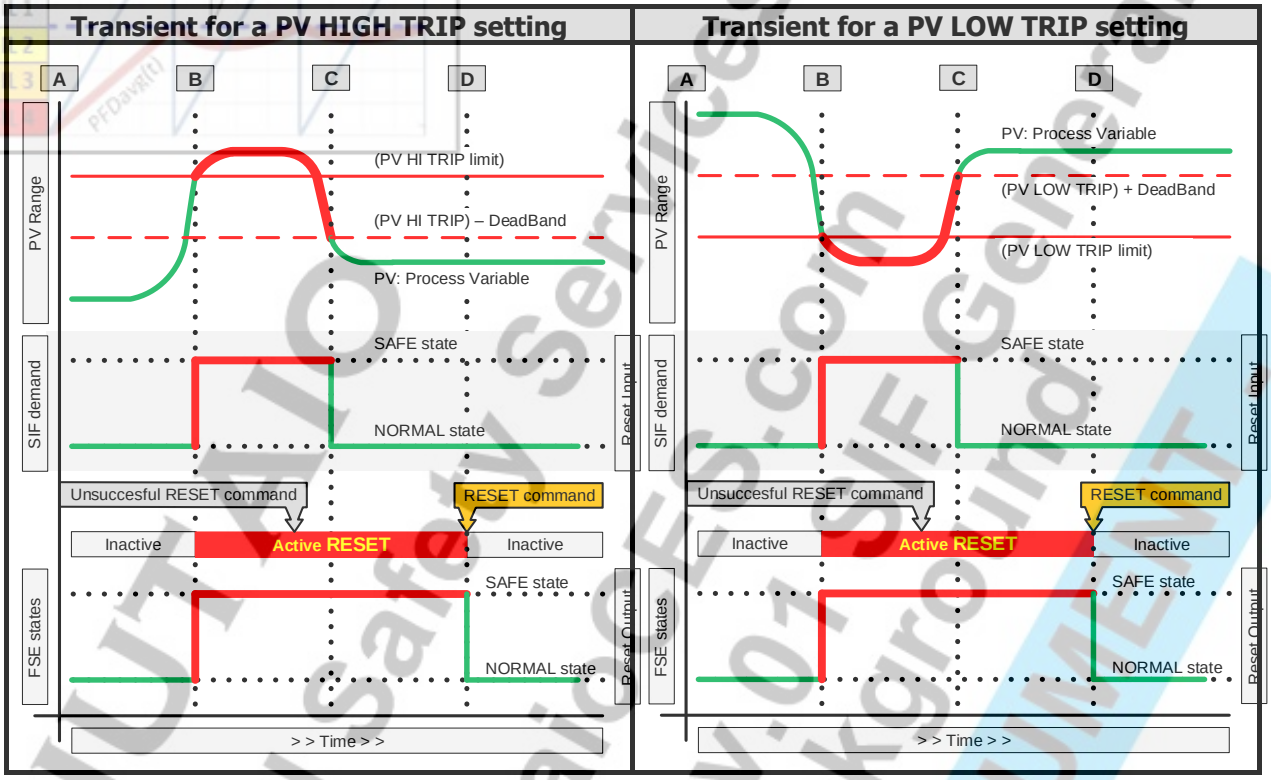
The "Reset Function" is a HARD REQUIREMENT for the design and installation of a SIF. The SIF design shall determine if the "Reset Function" will be implemented via a "Reset Logic", "Local Reset", or both.

5.1.2.1 Description

The purpose of a "Reset Logic" is to keep the FSE in the SAFE state after a SIF demand, even though the SIF "Initiator" is back in NORMAL state. In other words, the "Reset Logic" is activated after a SIF demand.

The "Reset Logic" input (SIF demand) is coming from the "SIF Decision Logic", or from the "Interlock Logic". Any of these logics can set the "Final Safety Element" (FSE) in SAFE state. BUT ONLY when both are in the NORMAL state, then it is possible to set the FSE in NORMAL state.

Figure 6 – Typical SIF and Rest function transients when a "SIF demand" occurs



The SIF "Initiator" can be an "Analogue" variable or a "Digital" variable.

The "Analogue" variable is typically a "Process Variable" (PV), or result of a calculation.

The "Digital" variable is a two(2) steps information. It can be a motor Running/Stopped status, a valve Opened/Closed status, etc. A "Digital" variable can be also the result of a "TRIP criterion".

Below paragraphs refer to Figure 6 for describing the "Reset function" behavior when the SIF "Initiator" is a "Process Variable" (PV, "Analogue" variable). The same description principles can be applied to for describing the "Reset function" behavior when the SIF "Initiator" is a "Digital" variable.

In Figure 6, initially @ time A the related SIF's "Process Variable" (PV) is in NORMAL state. In other words, PV value has not crossed the "TRIP limit" yet (High or Low).

@ time B when PV crosses the "TRIP limit", a "SIF demand" occurs that changes to SAFE state the SIF "Reset Function" and "Final Safety Element" (FSE), and the "Reset Logic" becomes in "Activate" condition.

Next, Operation and/or Maintenance personnel will solve the situation, and later it is possible to execute the actions to set PV back in NORMAL state.
@ time C, PV is back in NORMAL state NOT when it crosses back the "TRIP limit", in fact after it crosses back the "TRIP limit" in addition to a "Deadband" (or "Threshold"). BUT, the "Reset Logic" keeps the FSE in SAFE state, regardless the next states or conditions in the related machine, equipment or process plant.

A "RESET Command" **MUST BE EXECUTED** to make the "Reset Logic" to abandon the "Activated" condition, and to allow this logic input to determine the FSE's NORMAL or SAFE state.
The "RESET Command" **DOES NOT** work when the "SIF demand" is in progress. ONLY when "SIF demand" is over @ time D , the "RESET Command" is applied to set the FSE in NORMAL state.

5.1.2.2 Where to Implement

The "Reset Logic" can be implemented as a logic into the "Logic Solver", as a "Local Reset" via a mechanical device in the field, located at the "Trip device", FSE; or both.

The main purpose of the "Local Reset" is to force the "Field Operator" to go to the field to verify the good operation condition of the related machine/equipment/process plant before the manual "RESET Command" is applied to the "Local Reset" device.

NOTE: when the "Local Reset" is implemented, the "Reset Logic" should be implemented in the "Logic Solver" as well, in order to allow the Console Operator to conduct the machine, equipment or process plant startup in an order manner.

5.1.2.3 Automatic or Manual "RESET Command"

The "Local Reset" is Manual by nature.

The "Reset Logic" can be implemented in the "Logic Solver" with:

Automatic Reset	As soon as the "Reset Logic" input signal changes from SAFE to NORMAL state, and it remains in the NORMAL state, then the "Reset Logic" output AUTOMATICALLY follows this logic input condition.
Manual Reset	In the same condition as described above, BUT the "Reset Logic" output remains in the SAFE state. The Operator intervention is required to execute a "RESET Command" to make the "Reset Logic" output to follows this logic input condition.

Depending of the design, sometimes both Automatic/Manual Reset functionalities are implemented.

5.1.2.3.1 "RESET Command" location

Refer to section 3 for SIS's HMI general description.

The "Local Reset" shall be implemented in the field, normally at the "Trip Device".

The "RESET Command" for a "Reset Logic" can be implemented:

a) As a soft button in the SIS's HMI, Operation confirmation action.

b) As a physical push button in the safety panel.

NOTE: for both "RESET Command" implementations, "soft button" or "physical push button", the SIS's HMI shall show the "Reset Logic" input state. The Operator may execute the "RESET Command" at any time.

5.1.2.4 "RESET Command" implementation

The "RESET Command" shall be implemented as a PULSE signal. So, The Operator can apply the "RESET Command" as many times as required. This means:

a) For "Local Reset" and "Reset Logic" the Operator can apply the local "RESET Command" as needed, and

b) No history is required to maintain for a "RESET Command".

5.1.2.5 "Reset Function" condition after SIS Power up

When the SIS is power up the first time, or there is a SIS power shutdown, the "Reset Function" **MUST BE** in the "Activated" condition, in order to force the FSE in the SAFE state. This means:

a) For "Local Reset", the "Trip Device" is in the SAFE state, and

b) For "Reset Logic", on SIS powers up the it **MUST BE** set in the "Activated" condition.

5.1.2.6 "Reset Logic" HMI requirements

The Console Operator **MUST BE** able to monitor status of "Reset Logic" from DCS HMI.
If "RESET Command" is implemented via a soft-button, then Console Operator **MUST BE** able to command the "RESET Command" from DCS HMI as well.

Refer to "Appendix A" for further details.

5.1.3 Interlock Logic

The purpose of the "INTERLOCK logic" is to monitor some process variables or other logic/equipment conditions that **MUST** be satisfied to avoid undesirable states in the normal operation of an associated machine, equipment or process plant, and such conditions MUST be satisfied before Startup and/or the FSE is set in NORMAL state.

In other words, it is a function able to identify or detect undesirable operation conditions, in order to prevent the operation of a Machine, Equipment or Process Plant in such undesirable condition. In order to avoid Machine, Equipment or Process Plant in such undesirable condition, an INTERLOCK can:

a) Force the Machine, Equipment or Process Plant in the SAFE state, or
b) To allow predefined actions on the related Machine, Equipment or Process Plant to make it to react the desired startup or operation condition, or
c) The above both choices.

When the INTERLOCK is performing its function, it is "Enabled" and it sets the FSE in the SAFE state. An INTERLOCK always initiates a SIF demand.

When the undesirable condition disappears, the INTERLOCK condition changes to "Disabled", and it has no more effect over the related Machine, Equipment or Process Plant. BUT the "Reset Function" keeps the Machine, Equipment or Process Plant in the SAFE state.

The requirement of an INTERLOCK is determined by the Machine, Equipment or Process Plant Safeguarding and Startup procedure design. In some cases, an "INTERLOCK logic" is not required for a SIF implementation.

5.1.4 "Maintenance Override Switch" (MOS)

5.1.4.1 Description

The "Maintenance Override Switch" (MOS) is a facility that is used to administrate permission to apply online "Proof Test" or not to any SIF's device, or group of SIF devices.

If "Proof Test Logic" is implemented, MOS implementation is a **MUST**, else MOS is not required.

5.1.4.2 Constraints

Since the application of a "Proof Test" may deactivate (inhibit) partially or totally the safety purpose of a SIF, "Proof Test" **MUST BE** applied in an organized way in order to **DO NOT** compromise drastically the related machine, Equipment or process plant safety.

The philosophy to apply MOS **MUST BE** defined per project basis.

As main directives:

a) The maximum amount of activated MOS per "MOS Group" **MUST BE** defined. Once the number of current activated MOS is equal to the maximum allowed, other MOSs in the same "MOS Group" can be activated ONLY after any of the activated ones is set in the De-Activated state.

b) ONLY one MOS can be in the Activated state per SIF.

c) ALL SAFETY and PERMISSION procedures **MUST BE** completed and approved before activation of a MOS. A record **MUST BE** maintained for application of these procedures and "Proof Tests".

NOTE: Console Operator **MUST HAVE DENIED ACCESS** to any "Proof Test" commands where the related MOS is in the Deactivated condition.

d) Within a process plant, process unit, process train, equipment and SIF, "Proof Test" can be applied to ONLY one SIF device at the time. It means ONLY one MOS can be activated per functional or "Test Group".

e) When the related "Proof Test" is completed (finished), the related MOS **MUST BE** deactivated. Next, the "Proof Test" can be applied to another SIF device.

NOTE: a MOS **MUST NOT BE USED** as SIF override or bypass. If a functionality like this one is required, **IT MUST BE** implemented separately from the MOS.

f) Before "Proof Test" execution, the maximum time to allow a MOS to remain in the "Activated" state **MUST BE** equal or lesser than the MTTR (Mean Time To Restoration) of the associated SIF.

g) After "Proof Test" execution and if the "Proof Test" **WAS NOT** successful, the MOS time in "Activated" state can be extended by MRT time if the tested device can be repaired.

h) When a MOS is in "Activated" state, a timer shall start to account the remain time left for MOS activation. If the timer expires, then the MOS will be automatically deactivated.

This action:

- Will make the MOS input signal state to pass to all SIF elements downstream of the MOS, and

- IF MAINTENANCE activities have not finished within the MTTR time period, above action MAY initiate a TRIP.

NOTE: if more time is required to accomplish MAINTENANCE in a SIF device with activated MOS, the associated plant, equipment, unit, instrument, device **MUST BE** shutdown and set "Out Of Service" (OOS) to continue MAINTENACE activities.

5.1.4.3 Implementation

By default, a MOS shall be implemented for any SIF device where a "Proof Test" will be applied. If a "Proof Test" is applied to a group of devices together, then just one MOS shall be implemented for the whole group.

The SIF MOS **MUST BE** implemented in SIS. However, if the MOS administration is developed in DCS, this administration rules **MUST BE** honored in SIS as well. In other words, if it **IS NOT** allowed to activate a MOS in DCS, **IT MUST NOT BE** possible to activate the same MOS at SIS by using the SIS configuration tools.

PULSE signals between SIS and DCS shall be used to activate/deactivate a MOS. So, If DCS-SIS communication fails, MOS status at SIS **MUST** remain. When DCS-SIS communication is re-established, then DCS MOS status shall be initialized from SIS.

PULSE signals can be commanded by Soft-buttons or by Hard-buttons. The minimum requirements SHALL BE:

- For Hard-Buttons, to use Push Button with Light SPST momentary Push-to-make-and-Remain to Activate MOS. Console Operator SHALL push button again to De-Activate MOS.
- For Soft-Buttons, HMI push button with confirmation message.

MOS **SHALL NOT** inhibit device signal and/or statuses that are sent to SIS & DCS, for example Initiator process variable value.

Refer to section 5.1.4.4 for MOS implementation requirements for:

a) The SIF's Manual initiation signal (if it Manual input applies),
b) ALL SIF's Input MOS, and
c) ALL SIF's Output MOS.

5.1.4.4 MOS HMI requirements

The Console Operator **MUST BE** able to monitor status of, and to command each. "MOS Group" and each "MOS Device/Channel" from DCS HMI.

Refer to "Appendix A" for further details.

5.1.5 "Proof Test"

It shall be indicated for which SIF's Devices "Proof Test" can be applied, or not.
For each SIF's Device where "Proof Test" applies, indicate:

- Related MOS tag number.

- Description of the "Proof Test Logic" implementation: in DCS, in SIS, in external facility, or hybrid.
- Procedure to Lock/Unlock the "Proof Test".
- Procedure to operate the "Proof Test".

For each "Proof Test Logic", or portion of this logic, that is implemented in the "Logic Solver", a "MOS Logic" shall be implemented. Else, the "MOS Logic" is not required.

ALL SAFETY and PERMISSION procedures **MUST BE** completed and approved before operating any MOS and performing the related "Proof Test".

5.1.5.1 "Proof Test" LOG requirements

It is important to keep records of performed "Proof Test". This record should include, but it is not limited to:

- Tested Device (or Devices if an "Input Channel" is tested) Tag ID.
- MOS activation time.
- "Proof Test" start time.
- "Proof Test" completion time.
- Did test reach "Proof Test" target?
- MOS de-activation time.
- Was MOS de-activated before MTTR?

The "Proof Test" record can be maintained Manually in operation book notes, or it can be automated by a software application in DCS.

5.1.6 "Proof Test" on "Logic Solver"

It is not possible to apply "Proof Test" to a "Logic Solver" without affecting the performance of the SIF inside such solver.

ONLY when the related Machine, Equipment or process plant of ALL SIFs in the "Logic Solver" are shutdown, then a "Proof Test" can be executed in the "Logic Solver". The plant operation philosophy shall dictate if this test can include functional verification from input cards to output cards, or from "Initiators" to FSEs.

The "Logic Solver" implementation shall include the measures to allow MAINTENANCE personnel to isolate from Inputs and Outputs via software to perform "Proof Test". ALL SAFETY and PERMISSION procedures **MUST BE** completed and approved before executing this "Proof Test".

5.2 Additional SIF components per Input (Initiator)

5.2.1 Input Proof Test Logic

For the 1oo1 "Decision Logic", the "Input Proof Test Logic" can be applied only when all the following conditions are satisfied:

a) A control measurement of the same 1oo1 SIF "Initiator" exist, and this one can be used by the Console Operator to monitor the process safeguarding while the "Input MOS" is in the Activated state.

b) A Manual initiator push button is implemented to allow Console Operator to manually initiate the SIF action.

c) The SIF's "Safety Response Time" (SRT) is long enough to allow time to the Console Operator to react to manually initiate the SIF action on time to avoid the related SIF's Hazard.

For the XooN "Decision Logic" (X<=N), the "Input Proof Test Logic" **MUST BE** applied to each "Initiator".

The implementation of an "Input Proof Test" shall include:

- For an XooN "Decision Logic", the physical means to isolate the related "Initiator" from the SIF for independent test purposes.
- For an 1oo1 "Decision Logic", the physical means may not be required. "Initiator" isolation is handled by the "Input MOS". See next section 5.2.2.

By default, is assumed that "Proof Test" applies to each "Initiator" and the respective devices in the "Input Channel" at the same "Proof Test Period" and frequency. If this statement is FALSE, then for devices with different "Proof Test Period" and frequency:

1) Additional "Proof Test Logic" shall be implemented.
2) Additional physical means may be required to isolate the device test from the other devices.

"Proof Test Logic" functionality can be implemented in SIS, DCS, both, or in an external installation. And this logic depends on the Device type where the "Proof Test" is applied.

5.2.2 Input MOS

If "Input Proof Test Logic" is implemented for a Device, "Input MOS" implementation is a **MUST**, else "Input MOS" is not required.

The MOS implementation for each input of an XooN "Decision Logic" (X<=N) SHALL set in SAFE state the related "Input Proof Test Logic" output when the MOS is activated. In addition, the related "Decision Logic" (within the "SIF Decision Logic") shall be degraded. For example:

- From 2oo2 to: 1oo2,
- From 2oo3 to: 2oo2 or 1oo2 (project decision), etc.

The MOS implementation for 1oo1 "Decision Logic" **SHALL NOT** set the "Input Channel" in SAFE state. It **MUST** remain in NORMAL state while the "Input Channel" MOS is activated. The 1oo1 "Decision Logic" output **WILL BE** set in SAFE state by the "Proof Test Logic" when the "Proof Test" is in progress.

5.2.2.1 Input MOS integration with Diagnostic capabilities

If the "Initiator", "Input Channel" and "Logic Solver" can handle Diagnostics, in order to identify "Detected Failures", the SIF implementation can take advantage of these capabilities in order to:

a) Warn Console Operator when a "Detected Failure" (Safe or Dangerous) occurs.
b) Avoid "Spurious Trips" when a "Safe Detected Failure" appears.
c) When it is required, make "Dangerous Detected Failures" to initiate a TRIP.
d) Improve the SIL rating and "Spurious Trip Rate" of the "Safety Instrumented Function" (SIF).

If 1oo1ND and/or (N-1)ooND "Decision Logic" can be implemented in the "Logic Solver", the related "Input Channel" MOS (or "Initiator") can be activated automatically. This MOS AUTOMATIC activation:

1) Shall be accounted as one of the activated MOS in the "MOS Group",
2) Can override the rule of maximum amount of activated MOS in the "MOS Group", and
3) It shall generate a warning for Console Operator.

In practice the 1oo1ND and (N-1)ooND "Decision Logic" can be performed ONLY by the "Logic Solver".

Nevertheless, when any of these "Decision Logics" is implemented in the "Logic Solver", the 1oo1D functionality can be inherited to other devices in the "Input Channel", if those devices can notify to the "Logic Solver" when a "Detected Failure" has been occurred. This is the way the above 1 to 3 benefits can be extended to the other Devices in the "Input Channel".

NAMUR NE 43 or "NAMUR sensor" are the simplest ways to allow an "Initiators" to communicate to "Logic Solver" when a "Detected Failure" has been occurred. ONLY in this case the single "Initiator" (or Device) can perform the 1oo1D "Decision Logic".

5.2.3 Manual Initiator

A "Manual Initiator" applies ONLY when the SIF's "Safety Response Time" (SRT) is long enough to allow time to the Console Operator to react to manually initiate the SIF action on time to avoid the related SIF's Hazard.

A "Manual Initiator" is implemented as an input to the "SIF Decision Logic", so this input can set the "Final Safety Element" (FSE) in the SAFE state, regardless the condition of the related SIF's "Initiators". Refer to Figure 5.

"Manual Initiator" function is normally provided to high-level safety trip in the plant safety hierarchy ONLY, **BUT** it may be provided lower plant safety hierarchy levels if SRT is low and the "Manual Initiator" is a design requirement.

A "Manual Initiator" can be implemented via a Soft-button or Hard-button. The minimum requirements SHALL BE:

- The button **MUST COMMAND** a two states Boolean signal to set the SIF in the SAFE or NORMAL states.
- For Hard-Buttons, to use Push Button with Light SPDT Push-to-make-and-Remain to initiate TRIP. Console Operator SHALL push button again to cancel manual TRIP command. By default, the Hard-Button NORMAL state shall correspond with the button position to keep the circuit in the "Energized" condition.
- For Soft-Buttons, HMI push button with confirmation message.

5.2.4 Manual Input MOS

The "Manual Input MOS" is connected to the "SIF Decision Logic" via the "Input MOS", and no "Proof Test Logic" is required.

The "Manual Input MOS" implementation logic shall keep this logic output in the NORMAL state when the "Manual Input MOS" is activated.

5.3 Additional SIF components per Output

5.3.1 Output Proof Test Logic

The "Output Proof Test Logic" implementation depends on the Device type and SIF design. It can be implemented in SIS, DCS, both or as a separate testing facility.

When the "Output Proof Test" is in progress, if the "Output Proof Test Logic" input changes from NORMAL to SAFE state and remains, the logic output **MUST CHANGE** to SAFE state as well, regardless in which state or condition the "Output Proof Test" progress is.

In Figure 5 it is shown that the "Output Proof Test Logic" is located downstream the "Reset Logic". This means that a "Proof Test" can be performed even after a TRIP (or before Startup) when the SIF is setting the FSE in the SAFE state. This is done in this way because for some designs just before startup is the right time to perform "Proof Test" in the "Output Channels".

NOTE: it is the Operation responsibility to authorize, or not, the activity to perform "Proof Test" in the "Output Channel".

Regardless of the location(s) where, and how, an "Output Proof Test Logic" is implemented, if the "Logic Solver" sets the "Final Safety Element" (FSE) in SAFE state, this action MUST BE executed, regardless the "Proof Test" in progress, and the related MOS condition.

5.3.2 Output MOS

If "Output Proof Test Logic" is implemented for a Device and this logic is partially or completely implemented in the "Logic Solver", then "Output MOS" implementation is a **MUST**, else "Output MOS" is not required.

APPENDIX A – DCS-SIS HMI typical tie-ins per SIF