

System Level Protection

The Watchdog Timer

Why a Watch Dog Timer

- Provides a Safety Mechanism Outside of a State Machine
- No Knowledge of What the Machine is Doing; Relies on Being Periodically Reset
 - Can not catch all errors
 - Can catch certain lockup states or sets of states
- Watch Dogs can be applied to hardware or software-based state machines

Sizing the Watch Dog Timer

Maximum Size

- Time to trigger + response must be less than the time to prevent the system damage.
 - Firing thrusters can deplete fuel
 - Off-sun pointing can result in discharged batteries
 - On-sun pointing can damage instruments (e.g., light down a telescope)
 - Loss of command and telemetry capabilities

Sizing the Watch Dog Timer

Minimum Size

- Adequate margin to prevent false triggering over mission lifetime
 - Software-based machine often have variable processing times.
 - Cache hit rates
 - Process switching
 - Responding to external events (interrupts)
 - Flexibility for a changing system
 - Software-based machines will be updated in flight
 - Reprogrammable FPGAs will be updated in flight

Sizing the Watch Dog Timer

Minimum Size (cont'd)

- Tolerance for credible failures
 - Double the expected time for trigger
 - An SEU can, worst-case, cut the duration of the timer by a factor of two
- Support test and diagnostic modes
 - Ground test
 - Diagnosis in flight

Watch Dog Timer

Credible Failures

- Expect SEUs, as discussed above
- Use a Trusted Power Supply
 - If non-radiation hardened devices are used, a Single Event Latchup (SEL) can bring down the power supply to the system, including the Watch Dog Timer, disabling the protection function.
 - A SEL may result in the entire subsystem being power-cycled.
 - Analyze power paths and various error scenarios carefully.

Watch Dog Timer

Useful Features

- Error Code Information
 - Store the fact that the WDT triggered in a safe memory
 - Power always on
 - Non-volatile memory
 - Enables operations to determine cause of a reset
- Error Count Information
 - Store as described above
 - Multiple triggers when out of ground contact
 - Use a 'saturating counter' design; no rollover

Watch Dog Timer

Useful Features (continued)

- Storage of Error Code Information
 - Use a non-volatile memory
 - Flash
 - EEPROM
 - Use a memory that is always powered
 - Solid state recorder (SSR)
 - Spacecraft computer to store information from an instrument
 - Small trusted memory that, along with the watch dog timer, is always powered.

Disabling the Watch Dog Timer

- Disable function, for a failed Watch Dog Timer
 - The WDT protection function can be a Single Point Failure
 - This failure of engineering equipment, which doesn't contribute to the mission, should not end the mission
 - Certain tests may require temporarily disabling the WDT
 - Difficult or unusual code patches

Enabling/Disabling Mechanism

- Use stored commands
 - Automatically resume protection after the test is over
 - Automatically resume protection if ground contact is lost
 - Automatically resume protection if something unforeseen goes wrong (e.g., new patch fails).
- Control with a hardware discrete command
 - Non-volatile storage mechanism (e.g., latching relay, trusted higher level system, etc.)

Enabled or Disabled?

What Should the Default Be?

- Often a good topic for debate.
- Enabled
 - Simple mechanism; far less likely to have failed than the system it is protecting
 - Provides protection when additional faults/resets happen when running autonomously
 - Gives the system a good chance at catching faults but may make certain recovery actions difficult

Enabled or Disabled? (cont'd)

What Should the Default Be?

- Disabled
 - Software people like this default.
 - Dealing with WDT resets can be awkward
 - Can leave the system open for mission ending failures.

Enabled or Disabled? (cont'd)

What Should the Default Be?

- Analyze and walk through ALL scenarios very carefully.