# CWE Detail – CWE-1209

## Description

The reserved bits in a hardware design are not disabled prior to production. Typically, reserved bits are used for future capabilities and should not support any functional logic in the design. However, designers might covertly use these bits to debug or further develop new capabilities in production hardware. Adversaries with access to these bits will write to them in hopes of compromising hardware state.

## Extended Description

Reserved bits are labeled as such so they can be allocated for a later purpose. They are not to do anything in the current design. However, designers might want to use these bits to debug or control/configure a future capability to help minimize time to market (TTM). If the logic being controlled by these bits is still enabled in production, an adversary could use the logic to induce unwanted/unsupported behavior in the hardware.

## Threat-Mapped Scoring

Score: 0.0

Priority: Unclassified

## Related Attack Patterns (CAPEC)

* CAPEC-121

## Modes of Introduction

**•** Architecture and Design: The Designer and Implementer have to make a conscious choice to do this

**•** Implementation: The Designer and Implementer have to make a conscious choice to do this

**•** Documentation: If documentation labels anything "for future use", "reserved", or the like, such labeling could indicate to an attacker a potential attack point

## Common Consequences

**•** Impact: Varies by Context — Notes: This type of weakness all depends on the capabilities of the logic being controlled or configured by the reserved bits.

## Potential Mitigations

**•** Architecture and Design: Include a feature to disable reserved bits. (Effectiveness: N/A)

**•** Integration: Any writes to these reserve bits are blocked (e.g., ignored, access-protected, etc.), or an exception can be asserted. (Effectiveness: N/A)

## Applicable Platforms

**•** None (Class: Not Language-Specific, Prevalence: Undetermined)

## Demonstrative Examples

**•** An adversary may perform writes to reserved address space in hopes of changing the behavior of the hardware. In the code above, the GPIO pin should remain low for normal operation. However, it can be asserted by accessing the reserved address space (0x0F). This may be a concern if the GPIO state is being used as an indicator of health (e.g. if asserted the hardware may respond by shutting down or resetting the system, which may not be the correct action the system should perform). In the code below, the condition "register\_address = 0X0F" is commented out, and a default is provided that will catch any values of register\_address not explicitly accounted for and take no action with regards to gpio\_out. This means that an attacker who is able to write 0X0F to register\_address will not enable any undocumented "features" in the process.