

When & Which

When & Which?

Comparison

Feature	<code>for</code> Loop	<code>while</code> Loop	<code>for-generate</code> Statement
Execution	Sequential	Sequential	Concurrent
Usage Location	Inside a <code>process</code>	Inside a <code>process</code>	Outside a <code>process</code>
Iteration	Fixed number of times	Repeats while a condition is <code>true</code>	Creates N physical copies
Purpose	Algorithmic tasks	Searching or Polling	Hardware Replication

Quick Guide

- Use a `for` loop when...
 - You need to repeat an action a **specific number of times**.
 - *Iterating through all the bits of a vector, initializing every address in a memory.*
- Use a `while` loop when...
 - You need to repeat an action **until a condition changes**, and you don't know how long that will take.
 - *Searching for the first occurrence of a value, waiting for a status flag to be set in a testbench.*
- Use a `for-generate` statement when...
 - You need to create **multiple, regular instances of hardware components** or concurrent statements.
 - *Building an N-bit register from N flip-flops, creating a chain of adders, connecting several identical modules to a bus.*

Remember, all loops have the same syntax rules, so you just need to remember the structure.
Good luck! :+1:

Revision #1

Created 2025-10-03 17:38:45 UTC by JD

Updated 2025-10-03 17:38:59 UTC by JD