## Aggressive GCC optimization:

- `–O3`  
- `–ftree-vectorize`  
- `–fprofile–generate`  
- `–fprofile–use`

- `–mtune=core2`  
- `–funroll–all–loops`  
- `–ffast–math`  
- `–fwhole–program`

- `–msse3`  
- `--march=amdfam10`

### Flag | What it does
---|---
`–combine` | Allows multiple source files to be optimized simultaneously
`–ffast–math` | Speeds up mathematical operations. **Warning:** results may be inexact; does not follow IEEE or ISO math specifications
`–fopenmp` | OpenMP 2.0 / libgomp support (GCC 4.2.0 and later)
`–ftree–vectorize` | Autovectorization (optimized in GCC 4.3 and later)
`–funroll–all–loops` | Maximizes processors' branch prediction features in looped code
`–fwhole–program` | Makes all global functions and variables static
`–mtune=core2` | Activates MMX, SSE, SSE2, 3DNow!, enhanced 3DNow!, and 64-bit instruction set extensions for AMD64 and AMD64-compatible processors
`–O3` | Aggressive optimization (see man page for full details)
`–msse3` | Enable SSE3 instructions (Next Generation AMD Opteron and later)
`–fprofile–generate` | These switches may be used with profiling
`–fprofile–use` | Tools such as AMD CodeAnalyst™ to guide optimization

---

**Optimize for AMD Phenom™** and code name “Barcelona” series processors*:

- `--msse4a`  
- `--mabm`  
- `--march=amdfam10`

* Barcelona support is available in AMD GCC RPMs on developer.amd.com and in the unreleased GCC 4.3 tree.

† the `--march` option provides superior performance to `–mtune`, but at the risk of introducing instructions not supported on non-target platforms.