**ChaCha20 Software Profiling Quantitative Breakdown (2MB Inputs, Preloaded)**

**Context:**

* Benchmark executed on pre-generated 2MB text inputs.
* Options [1], [2], [3] (Cipher Message, Cipher ASCII, Cipher Hex) tested.
* Total Time: **52.8 seconds**.
* Using cProfile, limited to top 20 functions.

**Quantitative Breakdown**

| **Category** | **Key Functions** | **Cumulative Time (s)** | **Percentage of Total** | **Comment** |
| --- | --- | --- | --- | --- |
| **Math / Encryption Core** | cipher(), chacha\_block(), quarter\_round(), int\_from\_string(), int\_to\_decimal\_string(), inner() | **48.7s** | **~92%** | ChaCha20 core logic and big integer XOR math |
| **ASCII/Text Conversions** | ascii\_to\_text(), text\_to\_ascii(), chr(), ord() | **1.7s** | ~3.2% | Format conversion between ASCII codes and text |
| **String Formatting** | to\_hex(), join(), <genexpr>, zfill() | **1.8s** | ~3.4% | Hex string formatting, internal joining operations |
| **Minor Overheads** | print(), load\_text\_from\_file() | **0.6s** | ~1% | Printing results, file I/O overhead |

**Important Observations**

* **Math/Encryption** is the **dominant bottleneck** (> 90%).
* **ASCII and String Handling** together account for only ~6%.
* **Minor overheads** like printing and file loading are negligible.

**Interpretation**

* **Priority for Hardware Acceleration:**
  + Focus on cipher(), chacha\_block(), and quarter\_round() implementations.
  + Also accelerate XOR operations and 512-bit math blocks internally.
* **Leave in Software:**
  + ASCII code conversion (ascii\_to\_text, text\_to\_ascii).
  + Hex formatting and random text preparation.
* **Outcome:**
  + Hardware accelerator will speed up **92% of total workload**, achieving maximum performance benefit.

**Conclusion**

"About 92% of the 52.8 seconds execution time comes purely from ChaCha20 encryption and math functions. ASCII conversions and text formatting account for a small ~6%, and can be safely left in software."