

先端計算アーキテクチャレポート課題

坪井正太郎 (101830245)

2021 年 1 月 29 日

1 概要

このレポートでは、最新の CPU として AMD の EPYC7742(Rome) を選択し、そのアーキテクチャをまとめた。

各性能について変化がある場合には、Zen/Zen+ アーキテクチャとの比較も行った。

2 仕様

表 1 仕様

Core	64
Thread	128
Base Frequency	2.25GHz
L3 Cache (Total)	256MB
PCI Express	PCIe 4.0 x128
Mem Channel	8

3 物理的なダイの構成

Zen2 アーキテクチャでは、複数のチップモジュールで 1 つのプロセッサを構成する。EPYC Rome は、8 つのコアが載ったダイ (Core Complex Dies(CCD)) とプロセッサの PCIe レーン、メモリ、コアチップレット同士の通信ハブとなるダイ (server IO die) から構成される。

このようにして IO とコアのダイを分ける設計には、以下のような利点があると考えた。

- ダイ製造の歩留まりが向上する。
 - 各ダイのサイズを小さくできることから、歩留まりの向上が見込める。
- 製品バリエーションを増やすことができる。
 - 実際に、EPYC と Ryzen は共通の CCX を持っている。コンシューマ向けとサーバ向けの IO ダイを用意して、CCX の接続数を変更するだけで製品特徴を変更できる。
- IO ダイと CPU ダイで独立して進歩できるようになる
 - 実際に、EPYC の CPU ダイのプロセスルールは 7nm だが、IO ダイでは 14nm が使われている。

- レイテンシの均一化

- Zen+ まででは、ダイの上にコアと IO を載せていたことで、同一ダイ間と異なるダイ間で行う通信のレイテンシに最大で倍の差がついていた。IO ダイを通すことで、レイテンシの均一化が図られ、どのような計算単位でも扱いやすくなった。

3.1 CCD の構成

EPYC の CCD は、2 つの Core Complex(CCX) で構成される。CCX は、L2 キャッシュを持つコアが 4 つ、16MB の L3 キャッシュを共有する形で実装される。

全体としては、EPYC(1 sIOD + 8 CCD(2 CCX(4 Core))) となる。

4 キャッシュ戦略

4.1 L1 キャッシュ

EPYC は L1 キャッシュとして、命令キャッシュとデータキャッシュを持つ。各キャッシュはそれぞれのコア内に配置される。

また、x86 命令はデコードされて内部命令の micro op となるが、その micro op を保持しておくキャッシュも搭載されている。内部命令に変換される理由は、スーパースカラやアウトオブオーダー実行を行うためであり、x86 プロセッサでは通常、縮小された内部命令セットを使用する。

4.1.1 命令キャッシュ

32KB, 8way のセットアソシアティブを採用している。Zen+ では 64KB, 4way のセットアソシアティブだったため、容量は少なくなっているが連想度を高めることでヒット時の速度を向上させる意図があると考えた。

4.1.2 micro op cache

32B の命令セットをデコードされて得られる内部命令を、2 つ分融合した内部命令をキャッシュとしておくモジュールである。容量は 4K 命令分であり、命令キューに対して 8 融合命令/サイクルで積む。これは、デコードが 4 内部命令/サイクルを出力できるので、融合された命令分を考慮して速度を合わせているのではないかと考えた。

4.1.3 データキャッシュ

命令キャッシュと同じく、データキャッシュも 32KB, 8way のセットアソシアティブであり、ライトバック方式を採用している。

4.2 L2 キャッシュ

L2 キャッシュは各コアに 512KB づつ割り当てられている。連想度は 8way であり、inclusive キャッシュである。inclusive キャッシュでは、上位キャッシュで持っているデータを、下位のキャッシュでも保持することがある。2 次キャッシュを inclusive にすることで、容量を犠牲にしてミスペナルティを削減しているの

ではないかと考えた。

4.3 L3 キャッシュ

L3 キャッシュは、各 CCX に対して 16MB, 16way で配置される。Zen+ と比較して、2 倍の容量となった。また、L3 キャッシュは L2 キャッシュから容量不足で追い出されたデータが保管される victim cache となる。これによって、L2 で犠牲になった容量分を確保していると考えた。

5 分岐予測

EPYC Rome では、分岐予測は 2 段階で行われる。L1 キャッシュを利用したパーセプトロンと、L2 までのキャッシュを利用した、より長い履歴から予測する TAgged GEometric history length predictor(TAGE) の 2 段階である。

TAGE はパーセプトロンによる予測よりも低速なので、パーセプトロンと異なる予測をした場合にのみ、TAGE の予測に基づいて実行し直される。2 段階にすることで、的中率と速度を両立することができると考えられる。

参考文献

- [1] T. Singh et al., "2.1 Zen 2: The AMD 7nm Energy-Efficient High-Performance x86-64 Microprocessor Core," 2020 IEEE International Solid- State Circuits Conference - (ISSCC), San Francisco, CA, USA, 2020, pp. 42-44, doi: 10.1109/ISSCC19947.2020.9063113. (<https://www.slideshare.net/AMD/zen-2-the-amd-7nm-energyefficient-highperformance-x8664-microprocessor-core>)
- [2] S. Naffziger, K. Lepak, M. Paraschou and M. Subramony, "2.2 AMD Chiplet Architecture for High-Performance Server and Desktop Products," 2020 IEEE International Solid- State Circuits Conference - (ISSCC), San Francisco, CA, USA, 2020, pp. 44-45, doi: 10.1109/ISSCC19947.2020.9063103. (<https://www.slideshare.net/AMD/amd-chiplet-architecture-for-highperformance-server-and-desktop-products>)
- [3] C. Ian, "AMD Zen2 Microarchitecture Analysis:Ryzen 3000 and EPYC Rome," (<https://www.anandtech.com/show/14525/amd-zen-2-microarchitecture-analysis-ryzen-3000-and-epyc-rome>)(2020-1-29 閲覧)