



ZOMI

达芬奇内核

Ascend



About

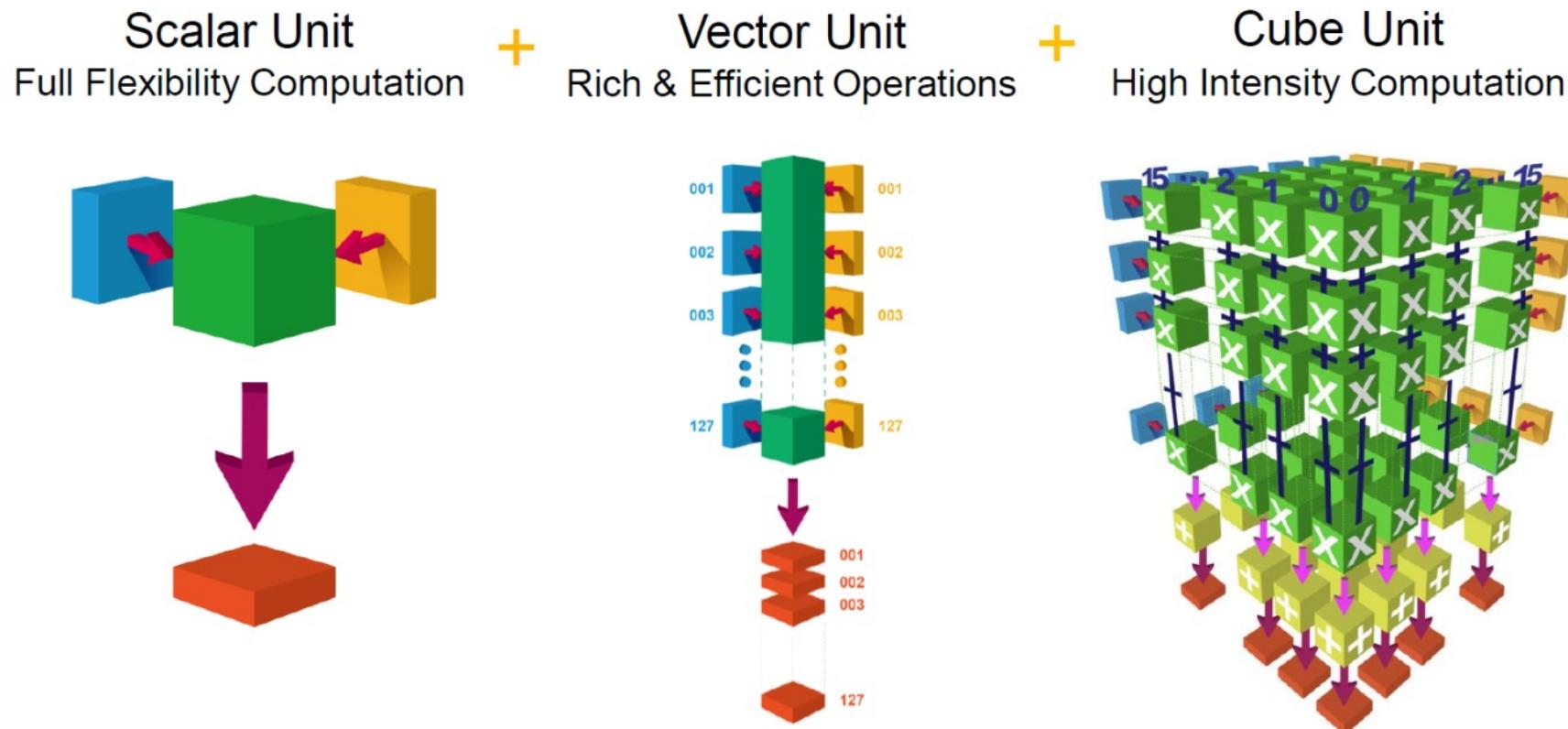
- **昇腾 SOC 架构:** 昇腾 310 芯片 - 昇腾 910 芯片
- **AICore 的灵魂:** 达芬奇架构内部细节
- **AICore 计算模式:** Vector 和 Cube 计算方法
- **昇腾服务器形态:** 节点互联拓扑



AI Core の 灵魂

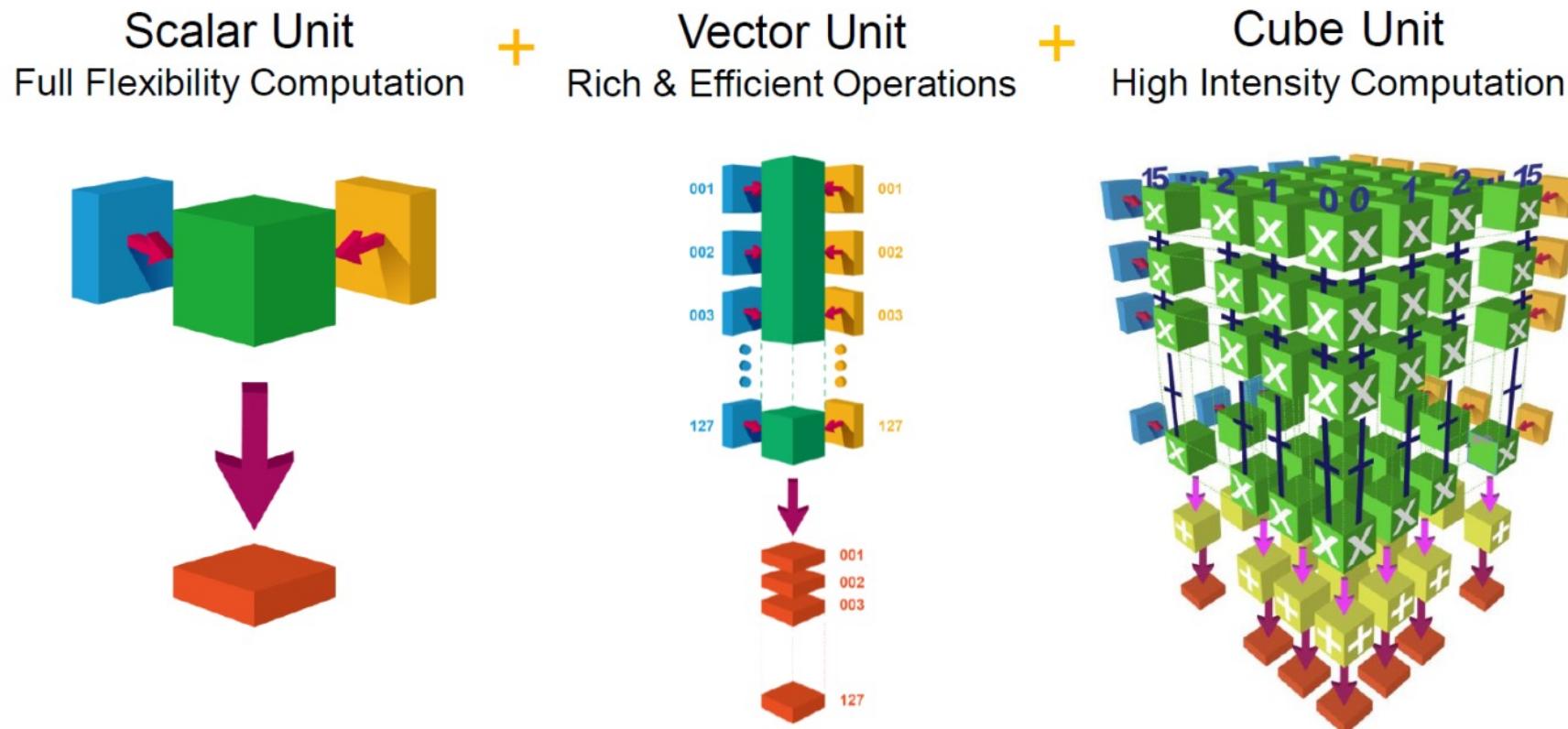
AI Core: 计算单元的加速原理

- **Cube 单元:** 绿色 X，代表两个数据间乘法运算，深蓝色 + 代表累加。浅蓝色矩阵 A，桔黄色矩阵 B，送入 Cube 单元后并行执行乘法运算，执行完后进行累加，放进地下累加器里。

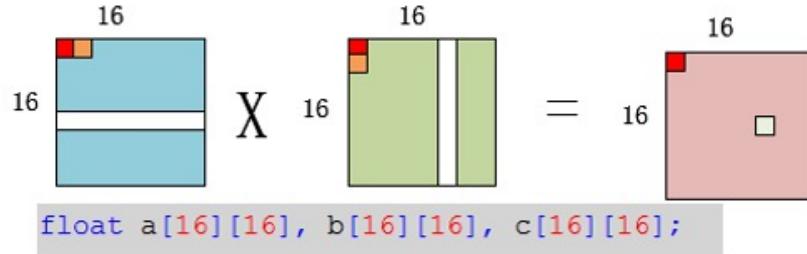


AI Core: 计算单元的加速原理

- **Cube 单元:** 目前 cube 但愿为 $16 \times 16 \times 16$, 通常矩阵乘中两矩阵很大, 因此数据是分块 (Tiling) 送入 Cube 单元中。每送完一块, 结果存放累加器, 最后得到结果。



AI Core: 计算单元 Cube Unit



CPU:

```
for(int i=0; i< 16; i++)
    for (int j=0; j<16; j++)
        for(int k=0; k<16; k++) {
            c[i][j] += a[i][k] * b[k][j];
        }
```

Cycle=16*16*16*2 = 8192
DataNum per cycle: Rd 2, Wr 1

Vector:

```
for(int i=0; i<16; i++)
    for ( int j =0; j<16; j++) {
        c[i][j] = a[i][:] *+ b[:,j]
    }
```

Cycle=16*16 = 256
DataNum per cycle: Rd 2*16, Wr 16

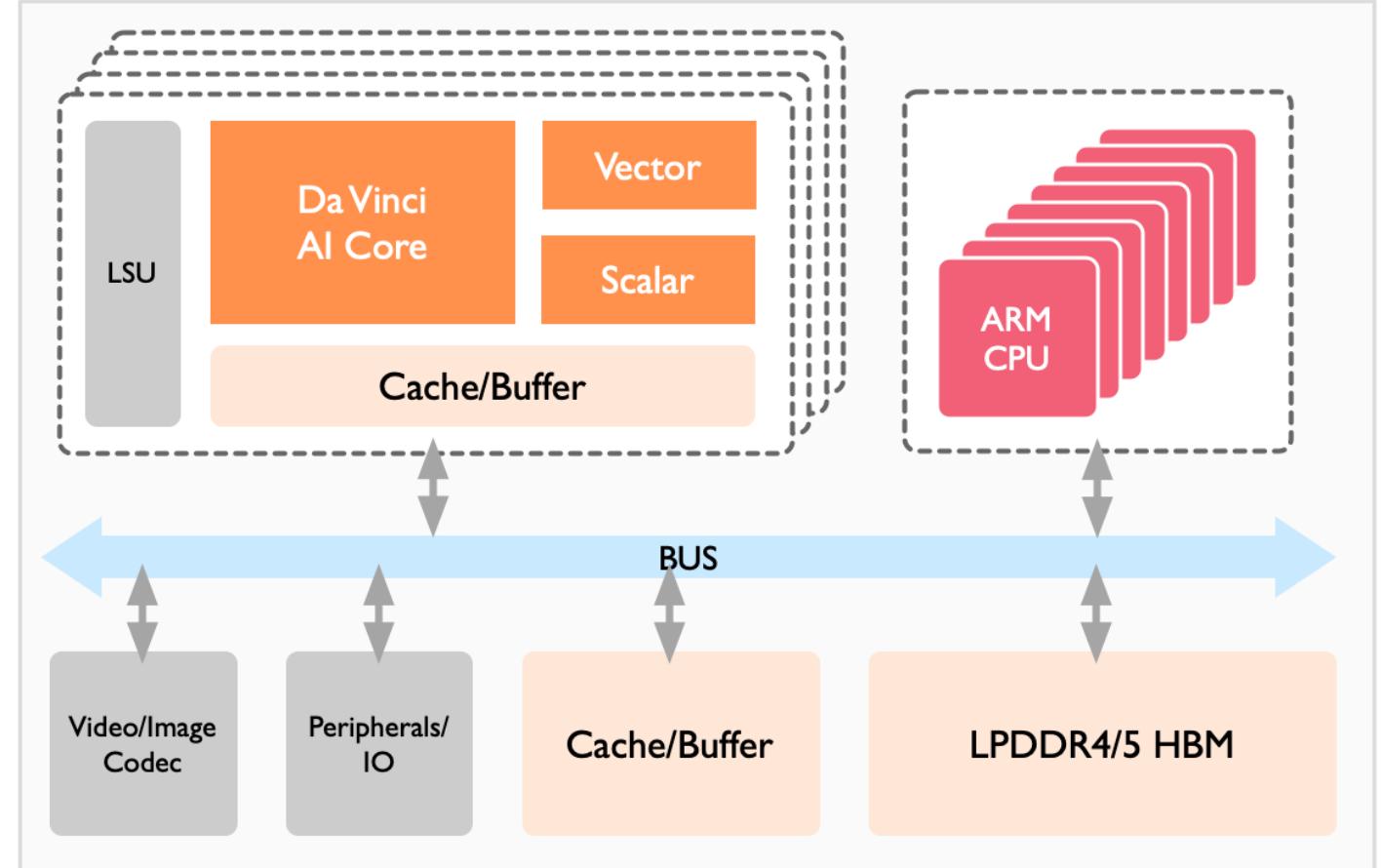
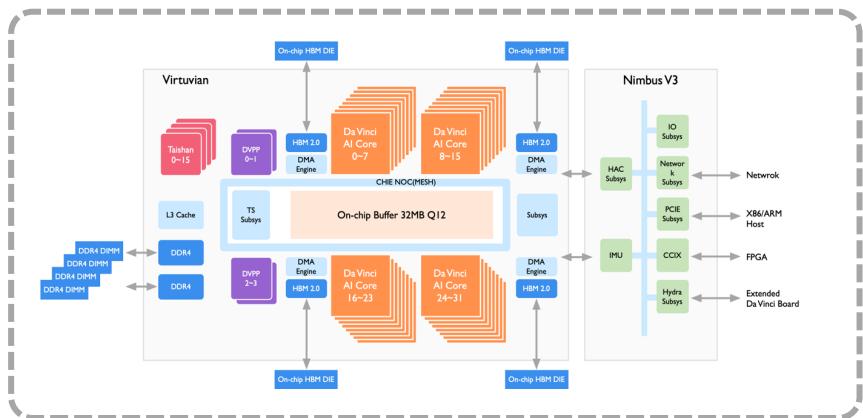
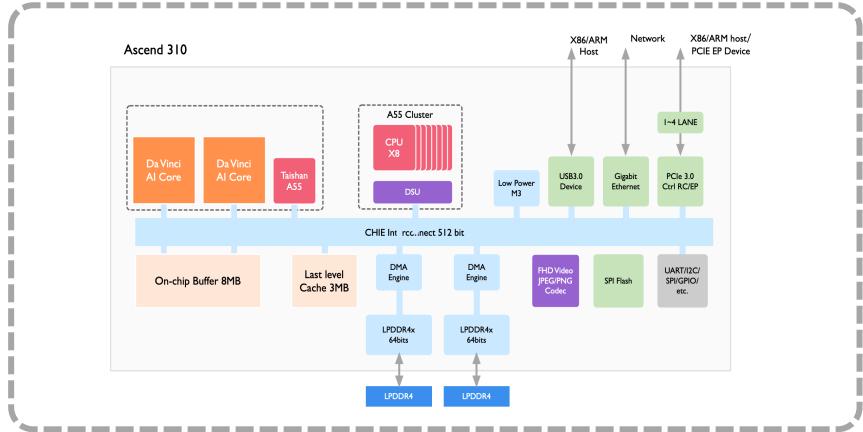
CUBE:

```
CUBE: c[:, :] = a[:, :] X b[:, :]
```

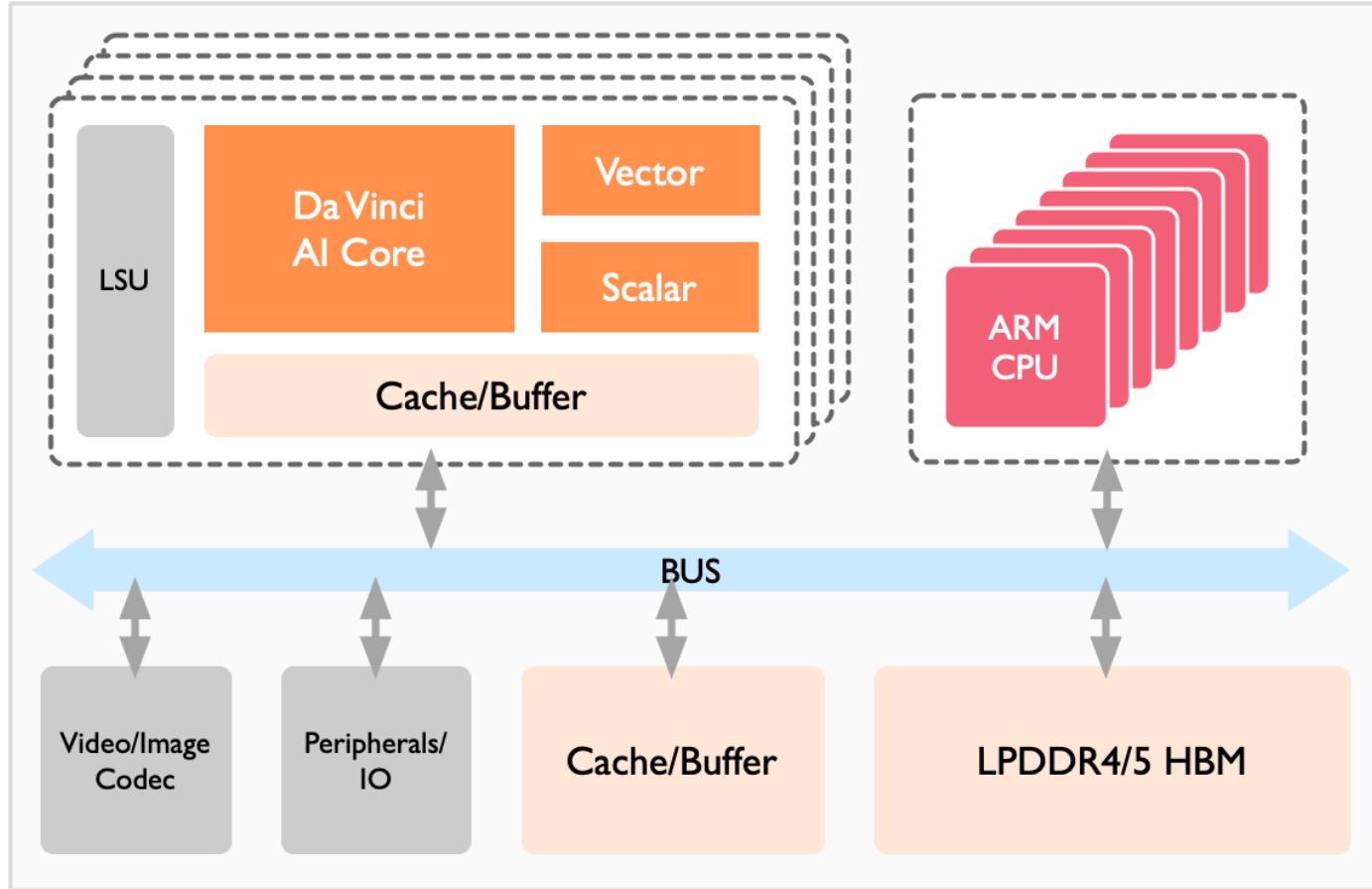
Cycle=1
DataNum per cycle: Rd 2*16*16
Wr:16*16

- 图例为一个矩阵 A 和另一个矩阵 B 间乘法运算 $C=A*B$ 。
- 在不同计算单元中实现该矩阵乘，其复杂度和运算效率差别很大。

AI Core 在昇腾处理器中的位置



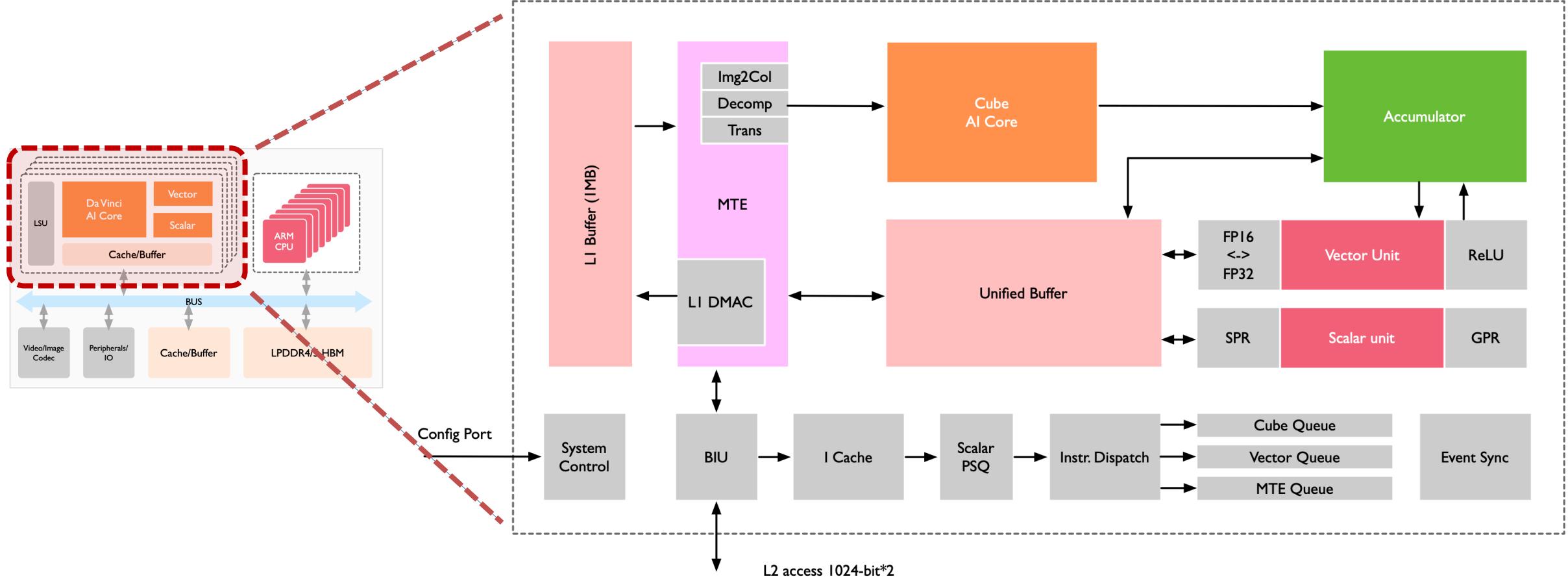
AI Core 在昇腾处理器中的位置



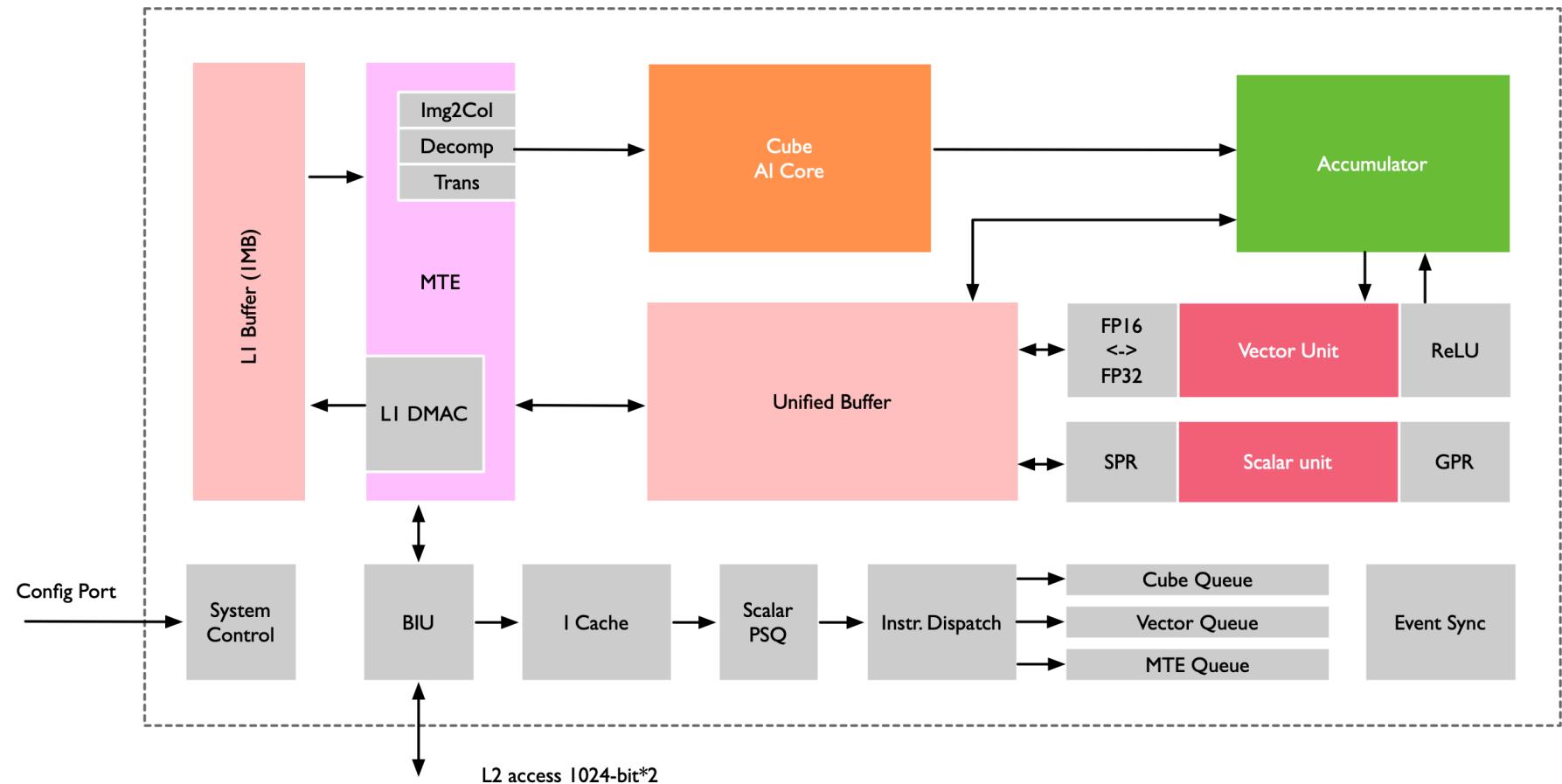
昇 AI 腾处理器：

- AI Core 是昇腾AI处理器计算核心，采用华为自研的达芬奇架构，通常也被叫做 DaVinci Core；
- 根据不同处理器版本，AI Core 里计算、存储和带宽资源有不同规格。

AI Core 架构总览



达芬奇架构主要部分



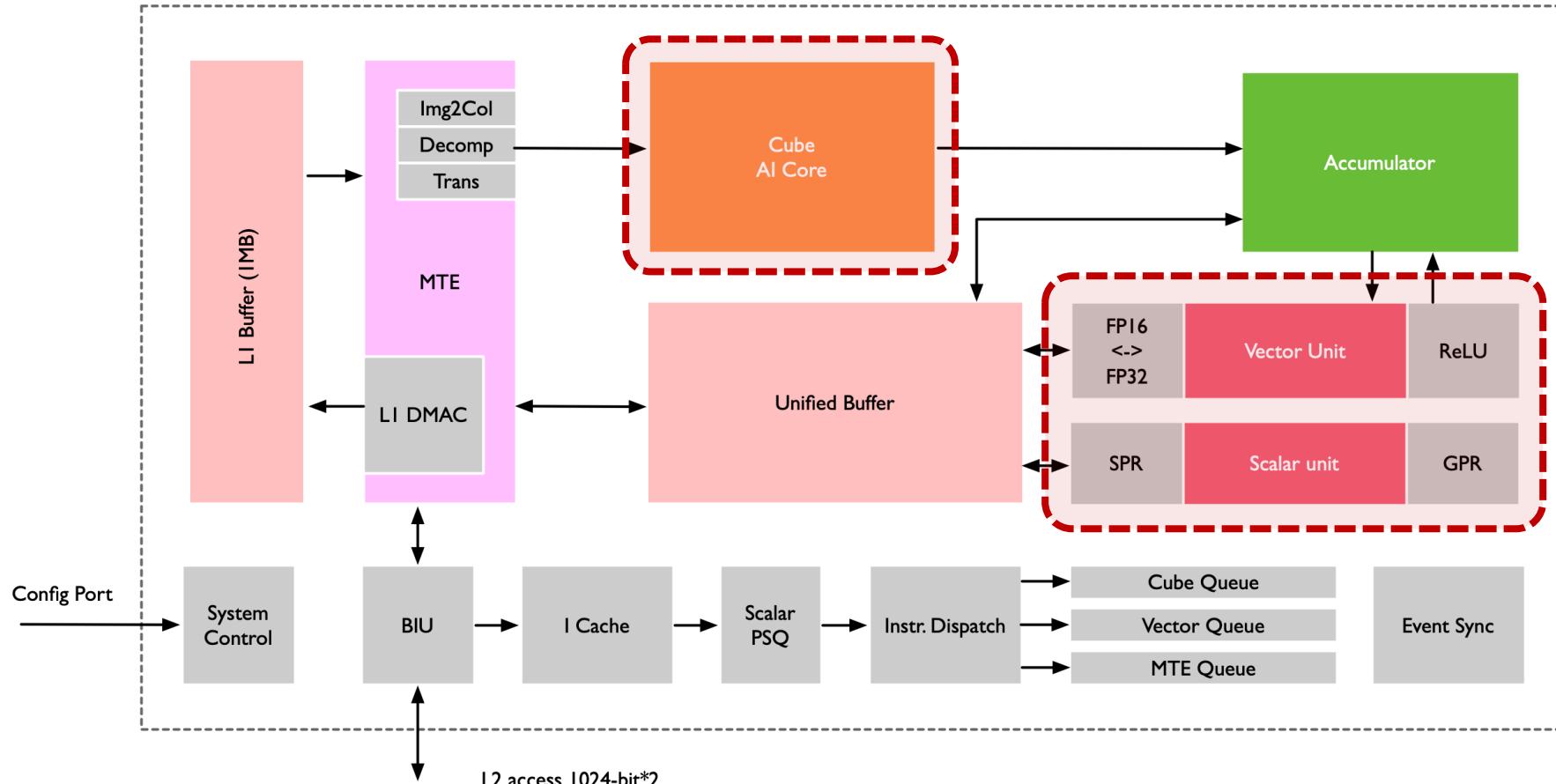
- 1. 计算单元:** 包含矩阵计算单元、向量计算单元、标量计算单元。
- 2. 存储系统:** AI Core 片上存储单元和相应数据通路构成存储系统。
- 3. 控制单元:** 计算过程提供指令控制，负责 AI Core运行。

2.1 AI Core

计算单元

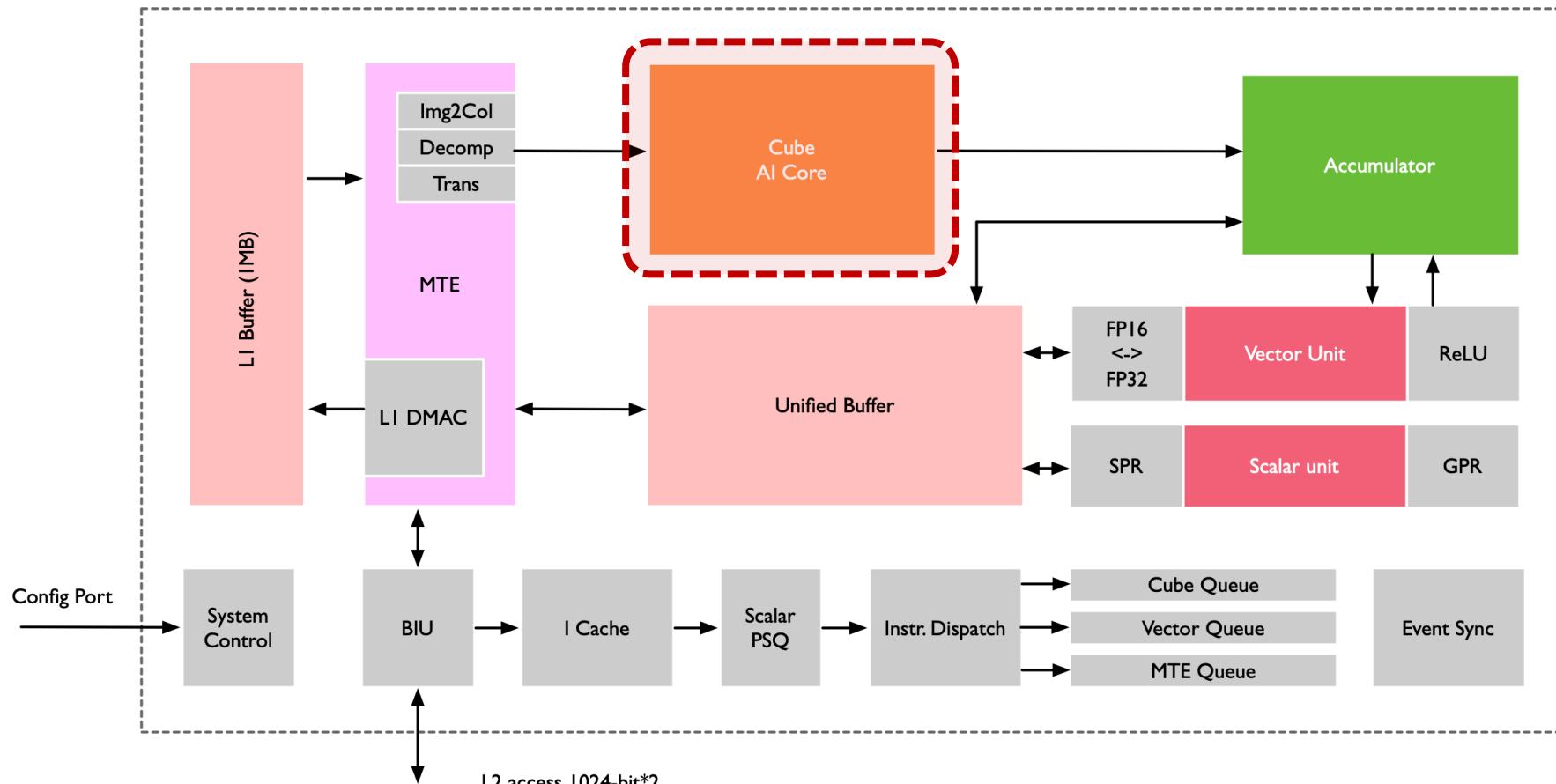
AI Core: 计算单元

- AI Core 中执行单元，包括：Cube Unit (矩阵计算单元) , Vector Unit (向量计算单元) 和 Scalar Unit (标量计算单元) , 完成 AI Core 中不同类型数据计算。



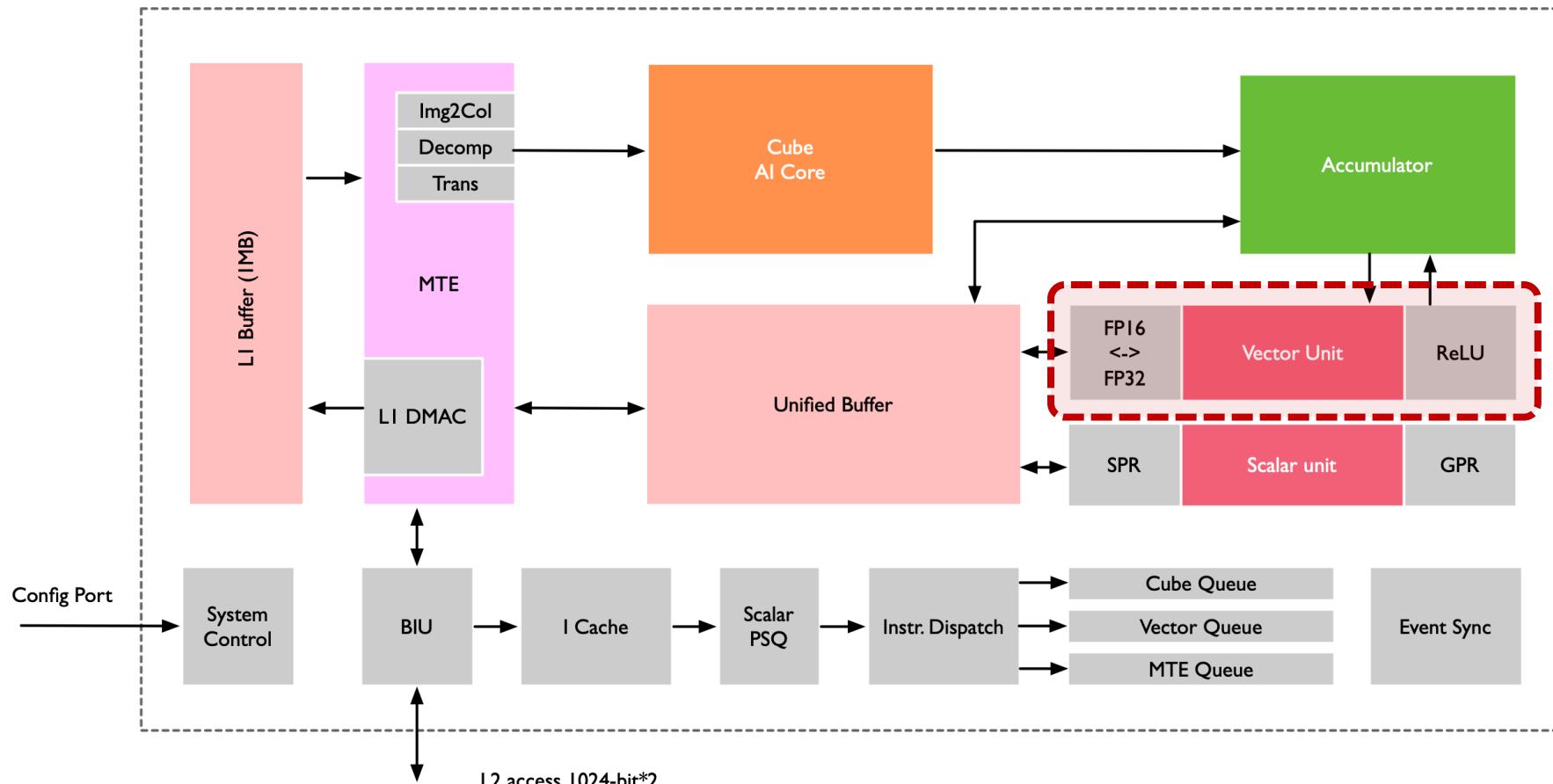
AI Core: 计算单元 Cube Unit

- 负责执行矩阵运算。Cube 每次执行可完成 fp16@16*16 与 16*16 矩阵乘，如 $C=A*B$ ，如果输入 int8，一次完成 16*32 与 32*16 矩阵乘，注意真正计算需要对 A/B/C 矩阵进行分块 Block。



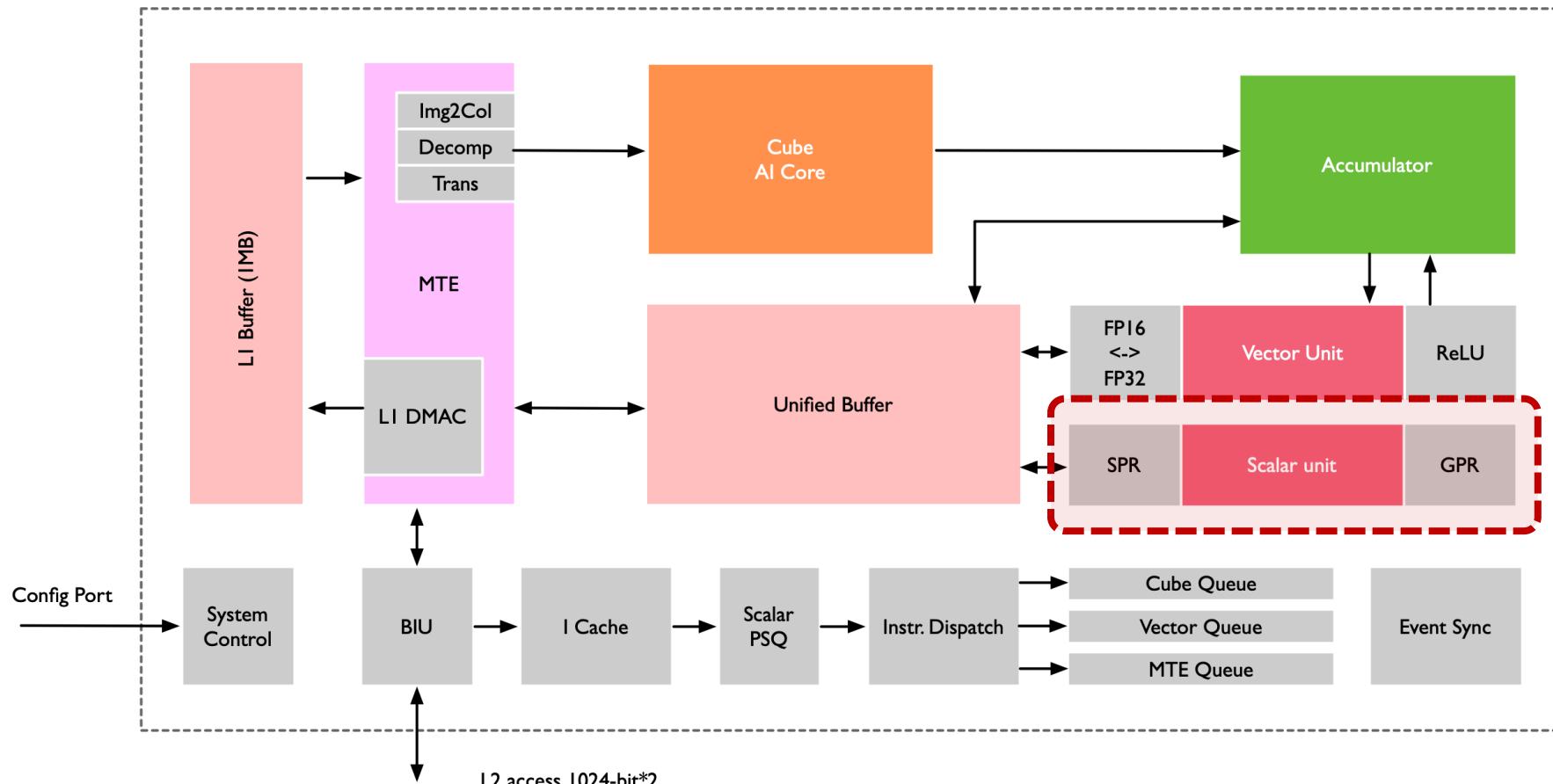
AI Core: 计算单元 Vector Unit

- 负责向量运算：算力低于 Cube，灵活度高（如数学中求倒数、平方根等），Vector 所有计算源数据以及目标数据都会存储在 Unified Buffer 中，并按 32Byte 对齐。



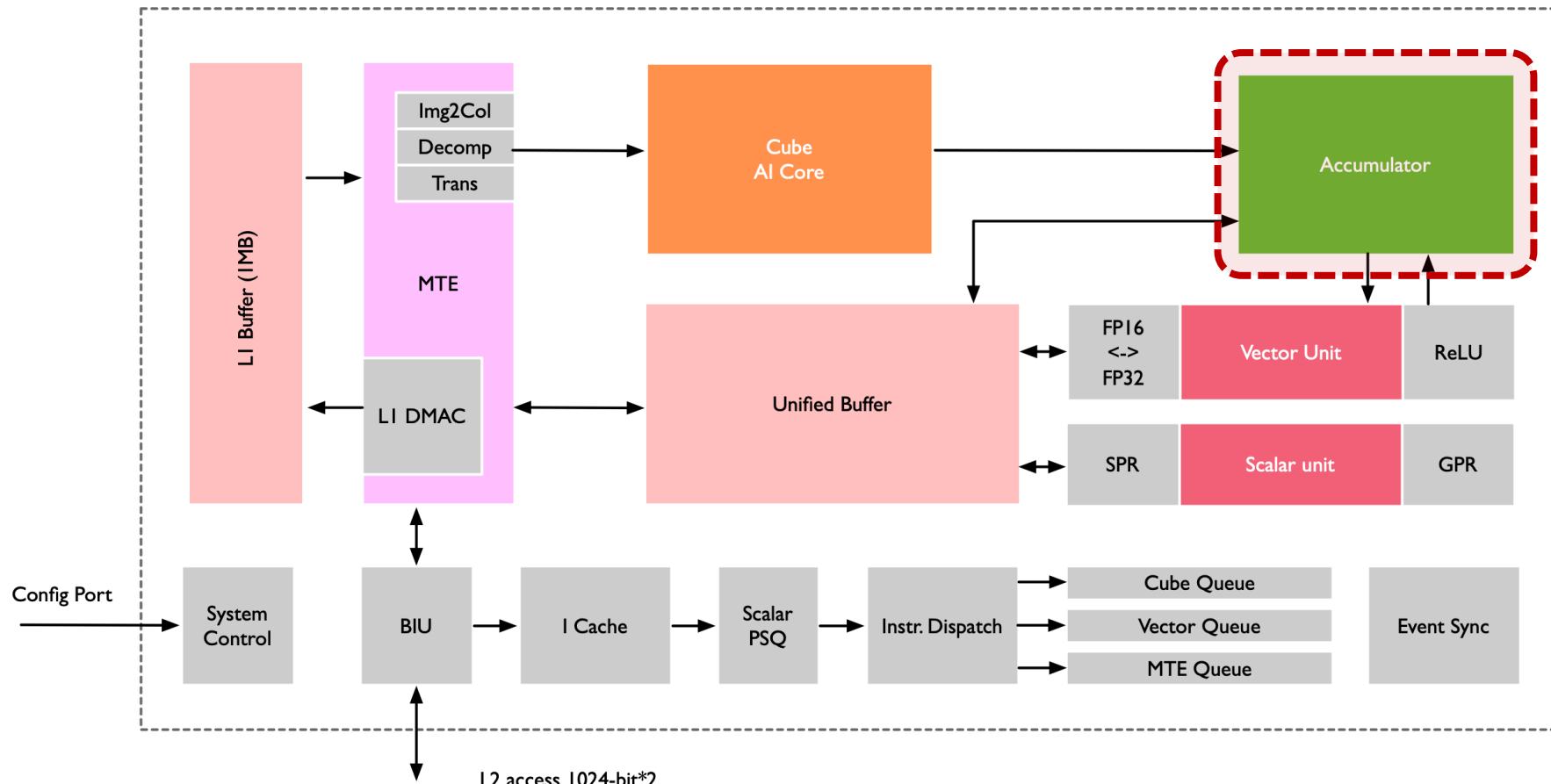
AI Core: 计算单元 Scalar Unit

- 负责各类型标量数据运算和程序流程控制：算力最低，功能上类比小核 CPU，完成整个程序循环控制、分支判断、Cube/Vector 等指令地址和参数计算以及基本算术运算等。



AI Core: 计算单元 Accumulator

- 累加器：把当前矩阵乘结果与上一次计算中间结果相加，可以用于完成卷积中加 bias 等操作。

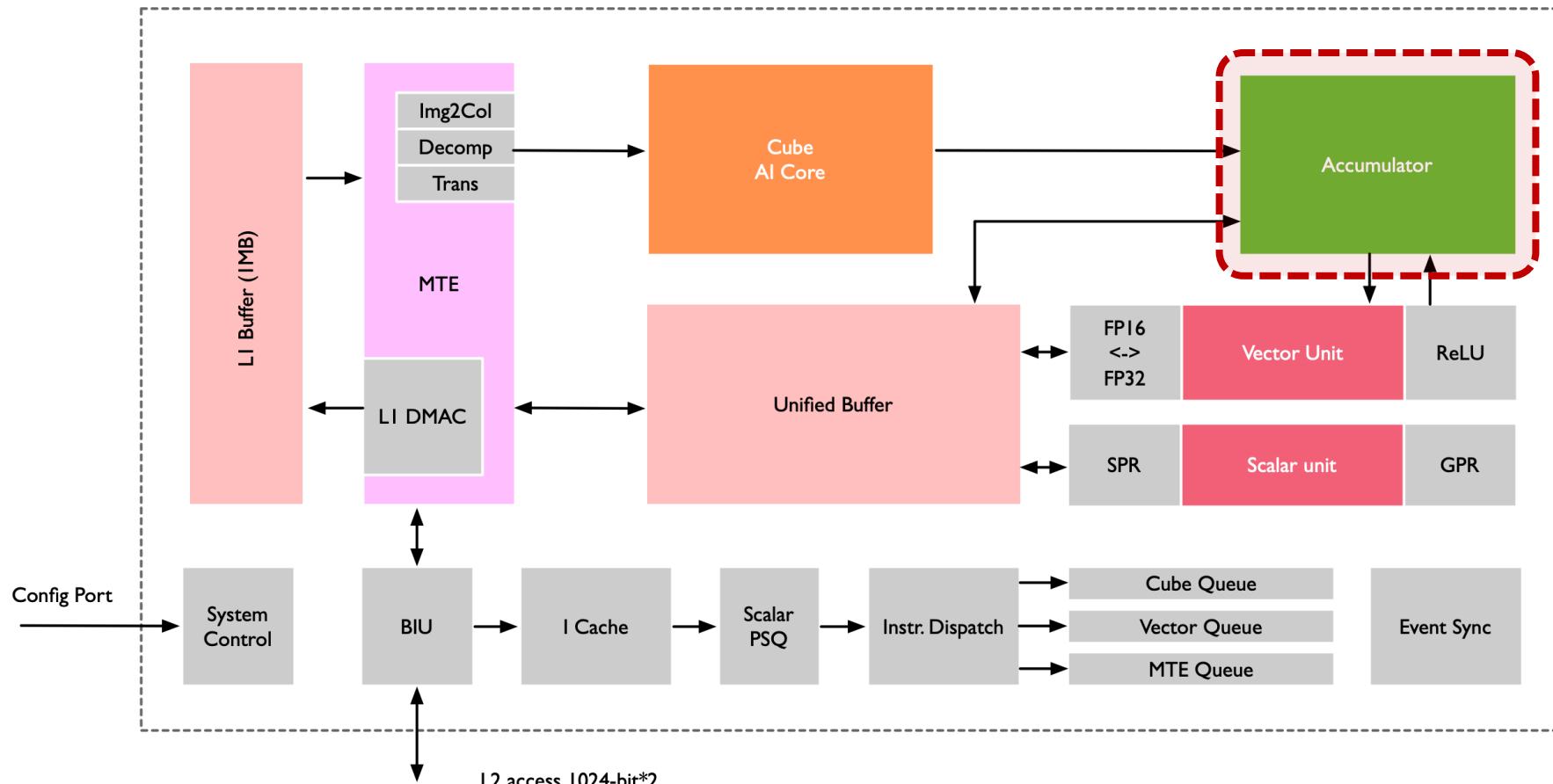


2.2 AI Core

存储单元

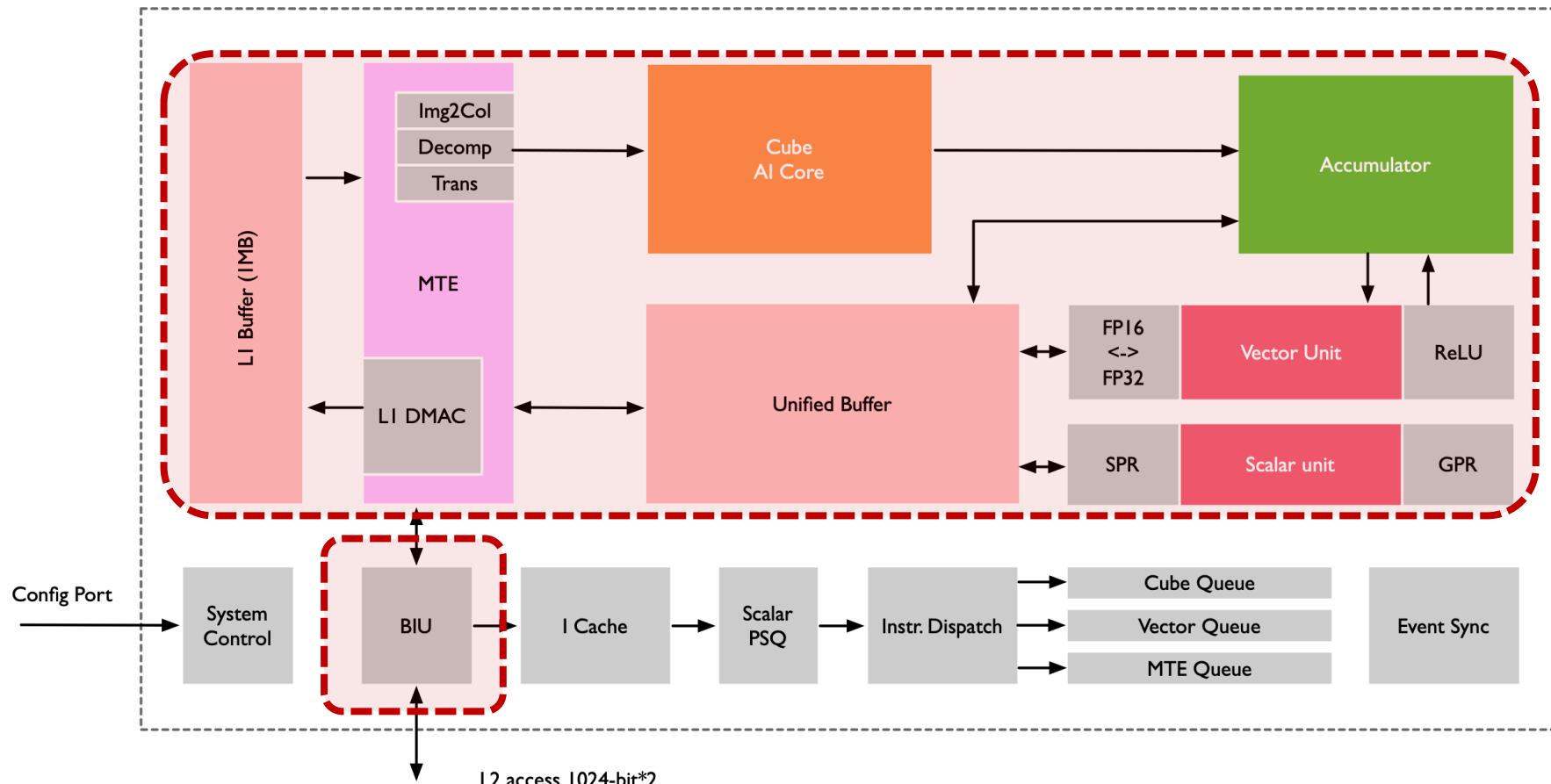
AI Core: 存储单元

- AI Core采用大容量片上缓冲区设计，通过增大片上缓存数据量来减少数据从片外搬运到 AI Core 中频次，从而降低数据搬运过程中所产生的功耗和时延，有效控制整体计算能耗和提升性能。

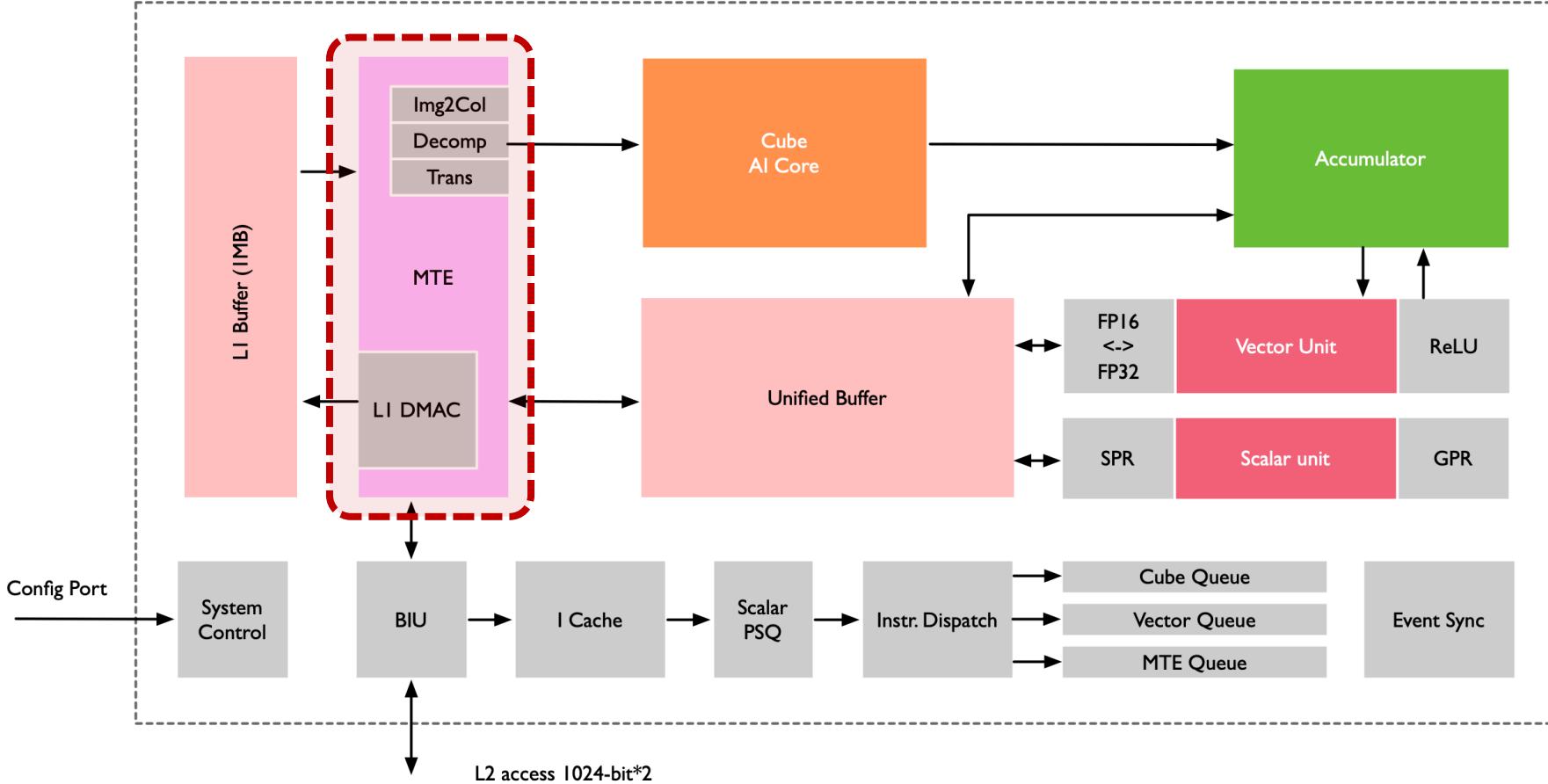


AI Core: 存储单元

- AI Core采用大容量片上缓冲区设计，通过增大片上缓存数据量来减少数据从片外搬运到 AI Core 中频次，从而降低数据搬运过程中所产生的功耗和时延，有效控制整体计算能耗和提升性能。



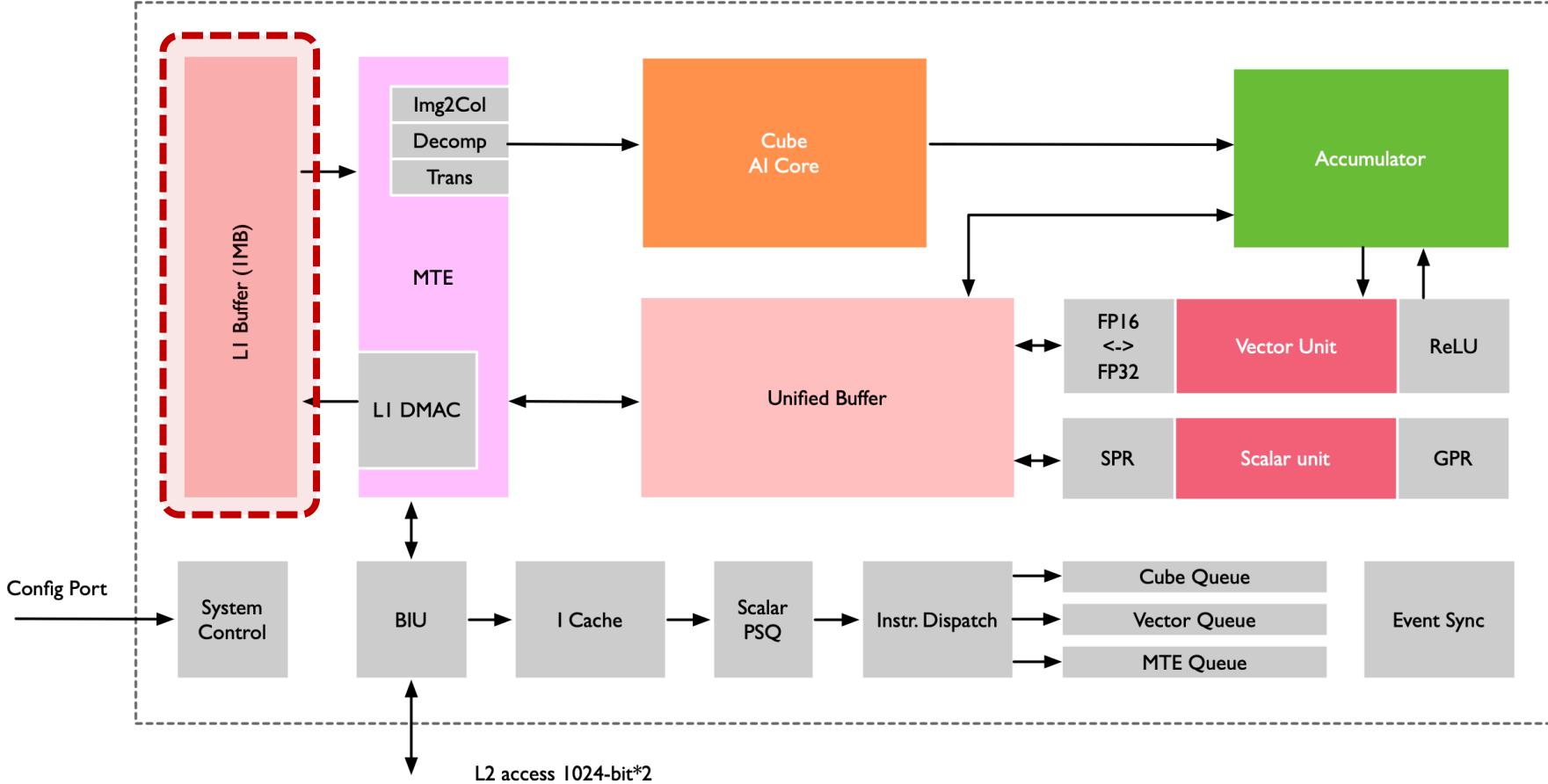
AI Core 存储单元：存储控制单元、缓冲区和寄存器组成



存储控制单元：

- 通过总线接口直接访问 AI Core 外更低层级缓存，也可直通DDR/HBM 访问内存。
- 设置存储转换单元，作为AI Core内部数据通路传输控制器，负责AI Core内部数据在不同缓冲区间读写管理，以及完成一系列的格式转换操作，如补零，Img2Col，转置、解压缩等。

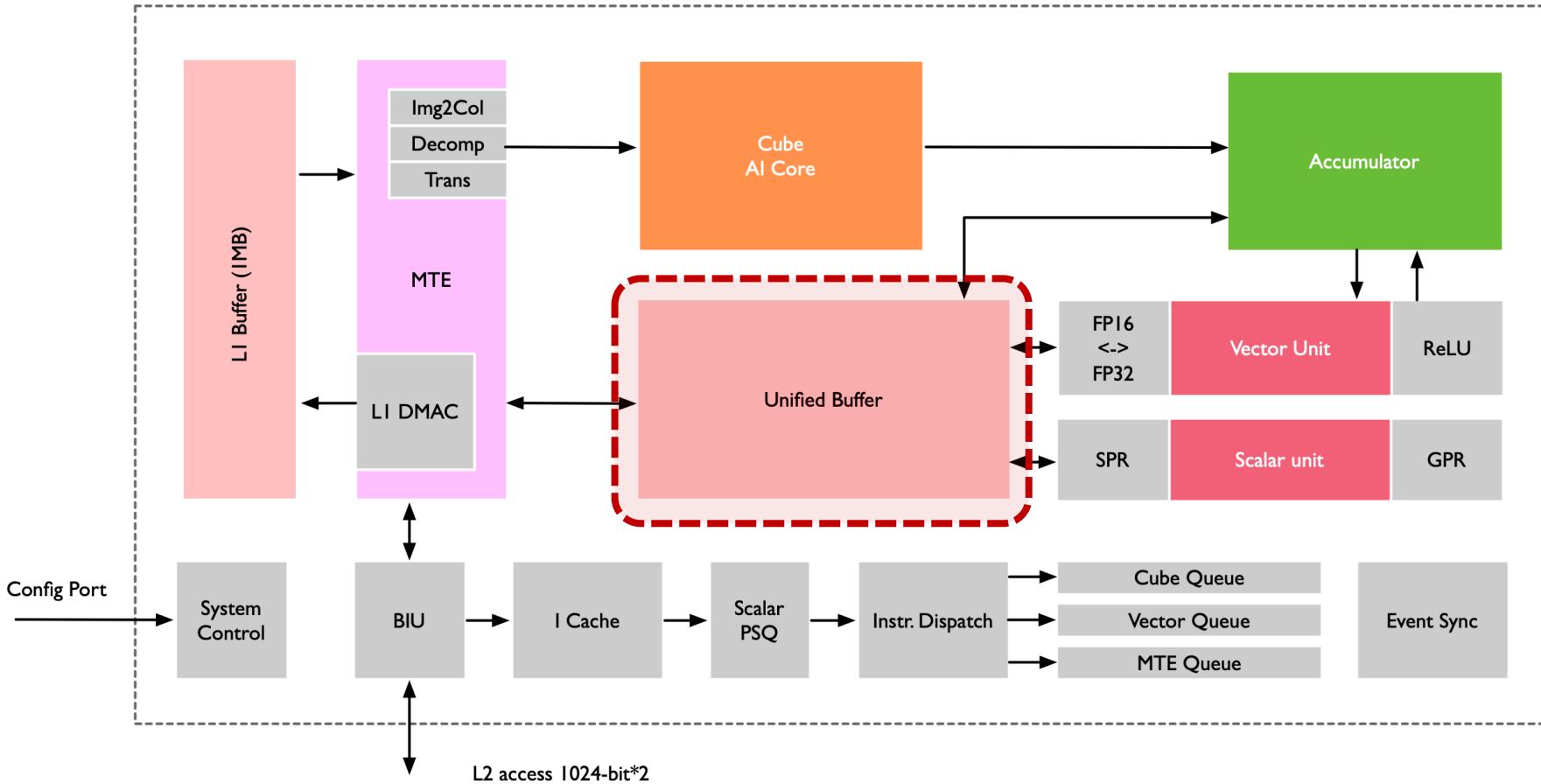
AI Core 存储单元：存储控制单元、缓冲区和寄存器组成



输入缓冲区：

- 用于暂存需频繁复用数据，不需要每次都通过总线接口到 AI Core 外部读取；
- 从而减少 BUS 上数据访问频次，同时降低总线数据拥堵风险；
- 实现节省功耗、提高IO、提升性能效果；

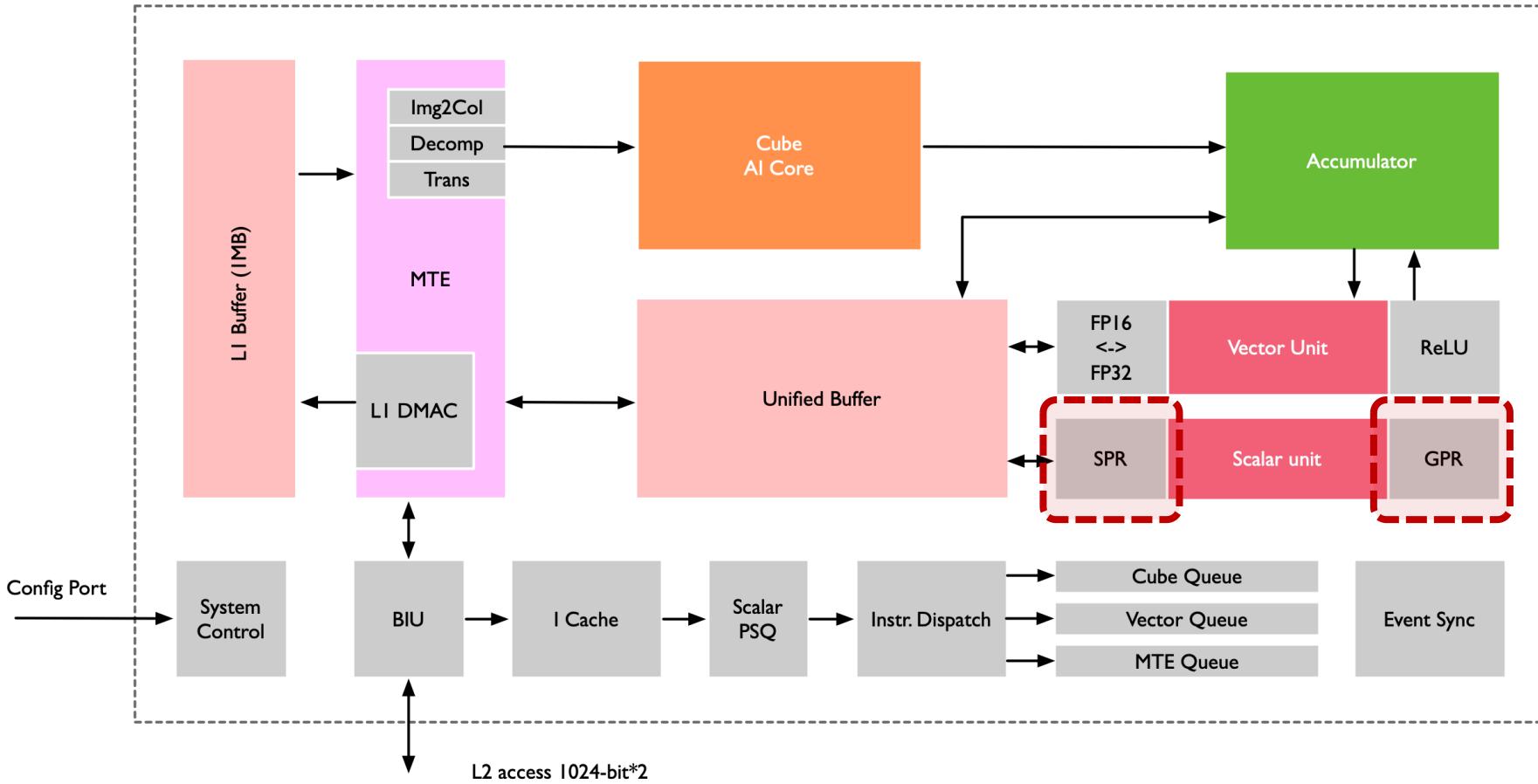
AI Core 存储单元：存储控制单元、缓冲区和寄存器组成



输出缓冲区：

- 用来存放神经网络中每层计算中间结果，从而在进入下一层计算时方便获取数据。
- 相比较 BUS 读取数据带宽低，延迟大，通过输出缓冲区可以极大提升计算效率；

AI Core 存储单元：存储控制单元、缓冲区和寄存器组成



寄存器：

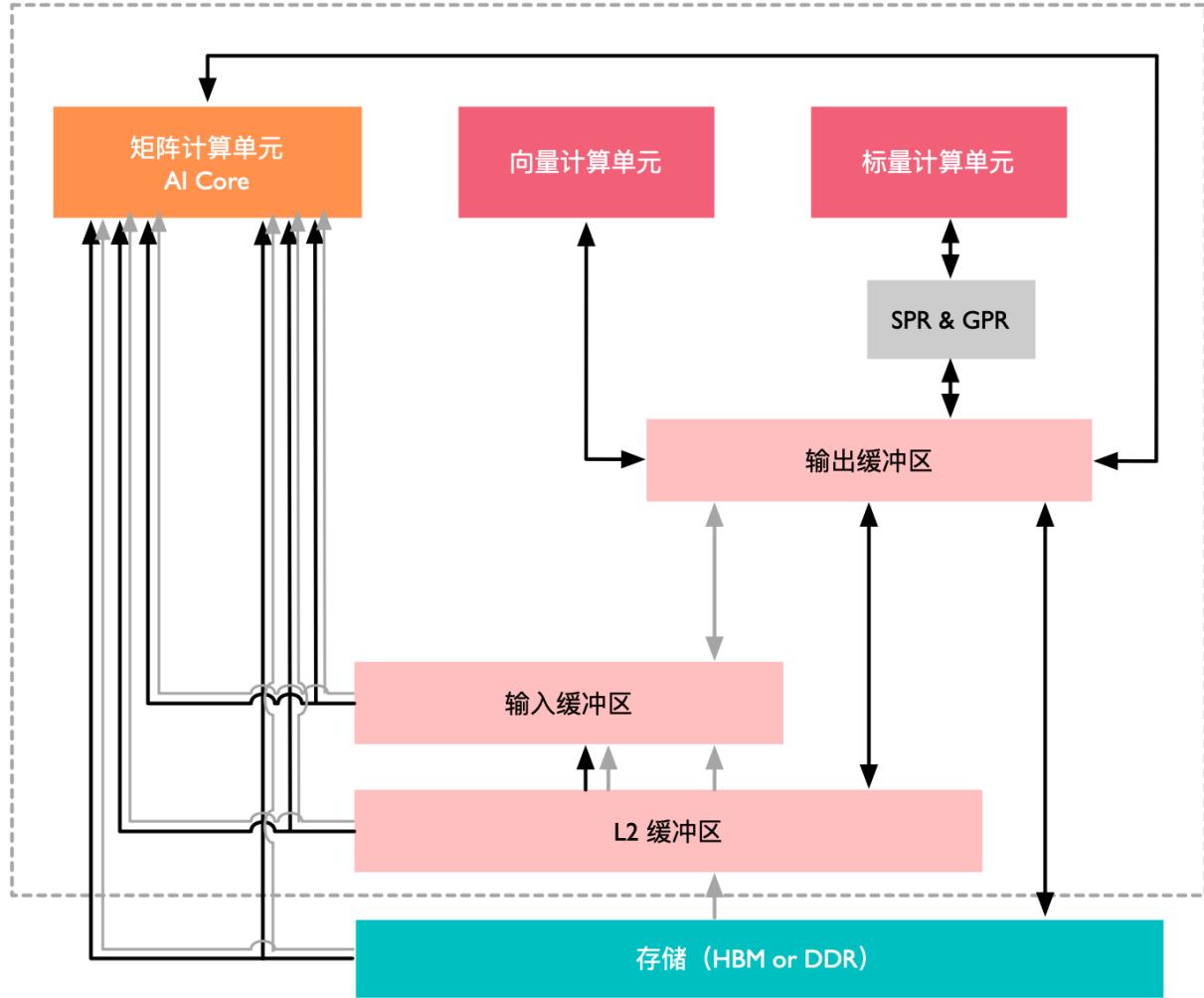
- 寄存器资源主要是标量计算单元在使用。

AI Core 存储单元：数据通路

AI Core 完成一次计算任务，数据在 AI Core 中流通路径：

- 达芬奇架构数据通路特点是多进单出，考虑到神经网络计算，输入数据种类多且数量大，可通过并行输入来提高数据流入效率。
- 与此相反，将多种输入数据处理完后只生成输出特征矩阵，数据种类相对单一，单输出数据通路，可以节约芯片硬件资源。

AI Core 存储单元：数据通路



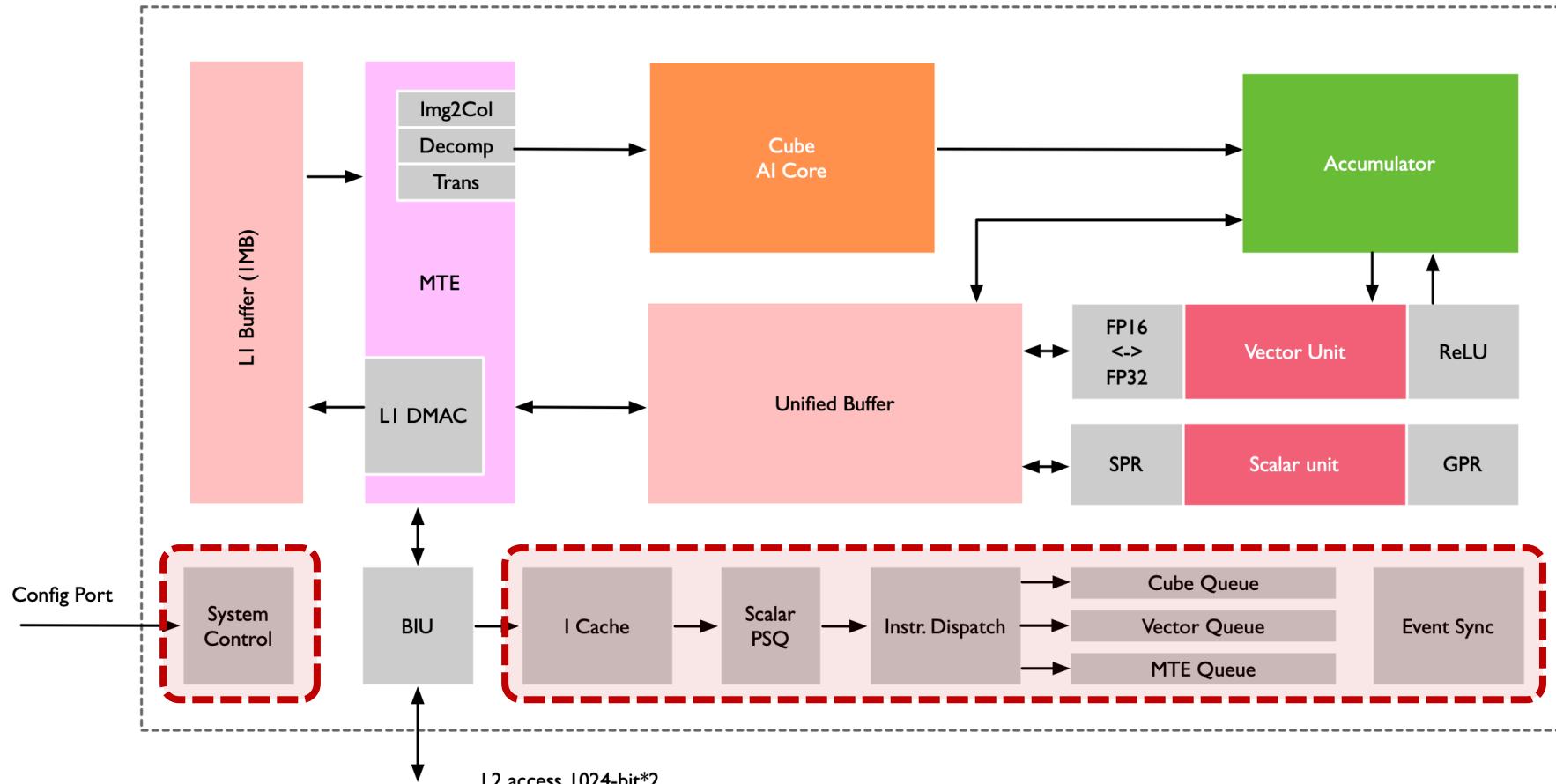
- 图中达芬奇架构中一个AI Core内完整的数据传输路径。包含DDR/HBM，以及L2缓冲器，这些都属于AI Core核外的数据存储系统。
- 图中其他各类型的数据缓冲器都属于核内存储系统，包括了多个通用和专用的寄存器。

2.3 AI Core

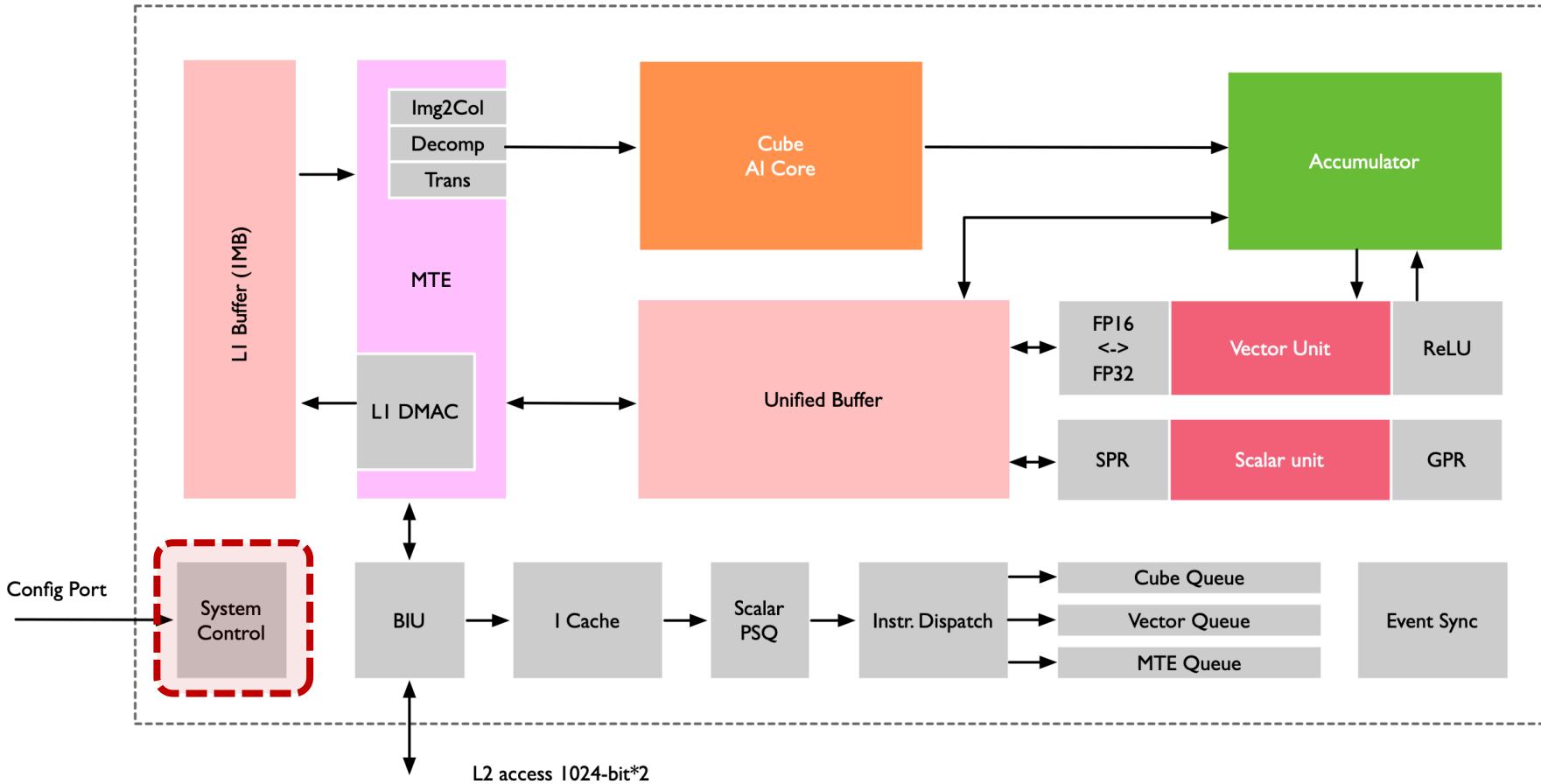
控制单元

AI Core: 存储单元

- 控制单元主要组成部分为系统控制模块、指令缓存、标量指令处理队列、指令发射模块、矩阵运算队列、向量运算队列、存储转换队列和事件同步模块。



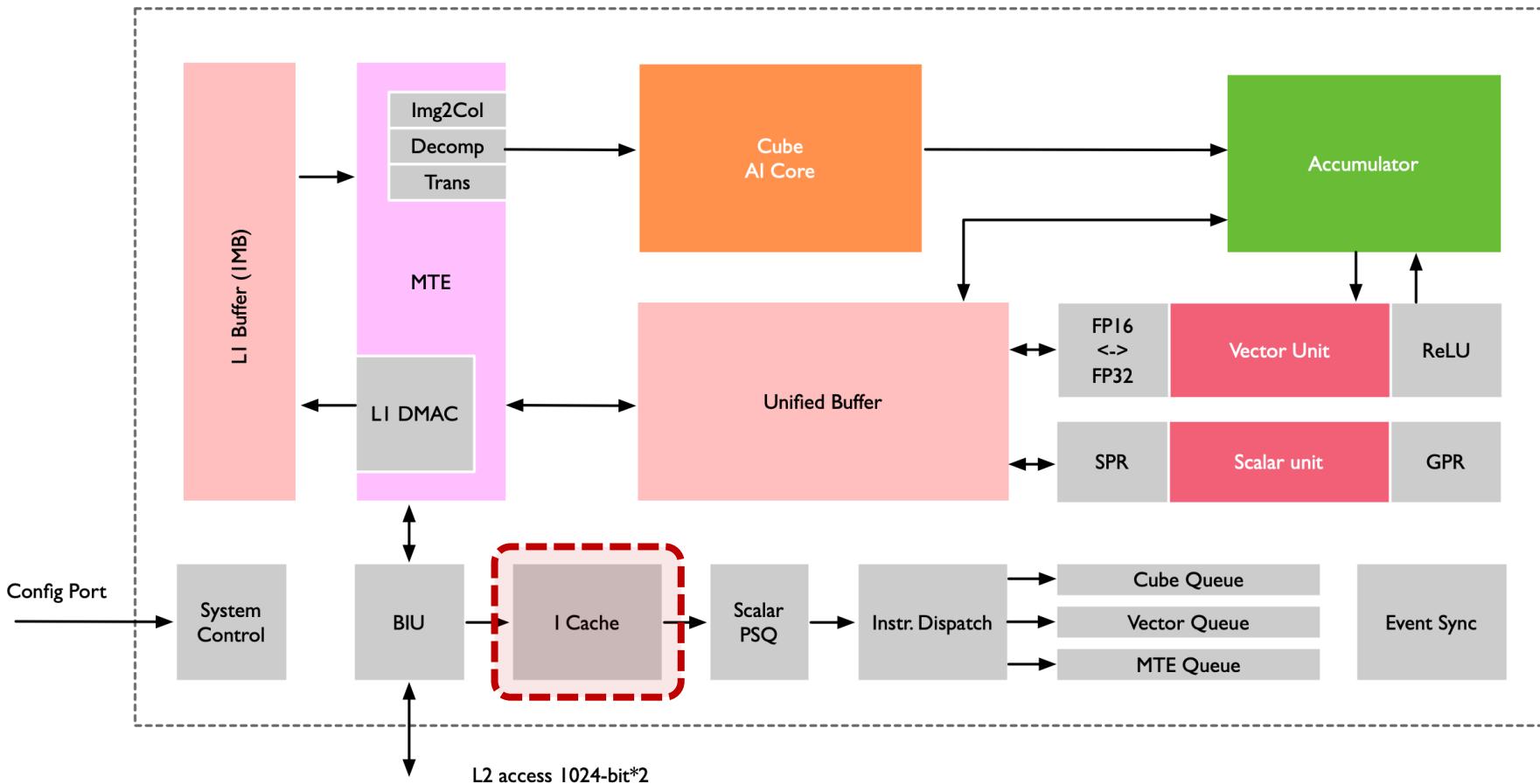
AI Core 存储单元：存储控制单元、缓冲区和寄存器组成



系统控制模块：

- 控制任务块 (AI Core最小任务粒度) 的执行进程，在任务块执行完成后，系统控制模块会进行中断处理和状态申报。如果执行过程出错，会把执行的错误状态报告给任务调度器；

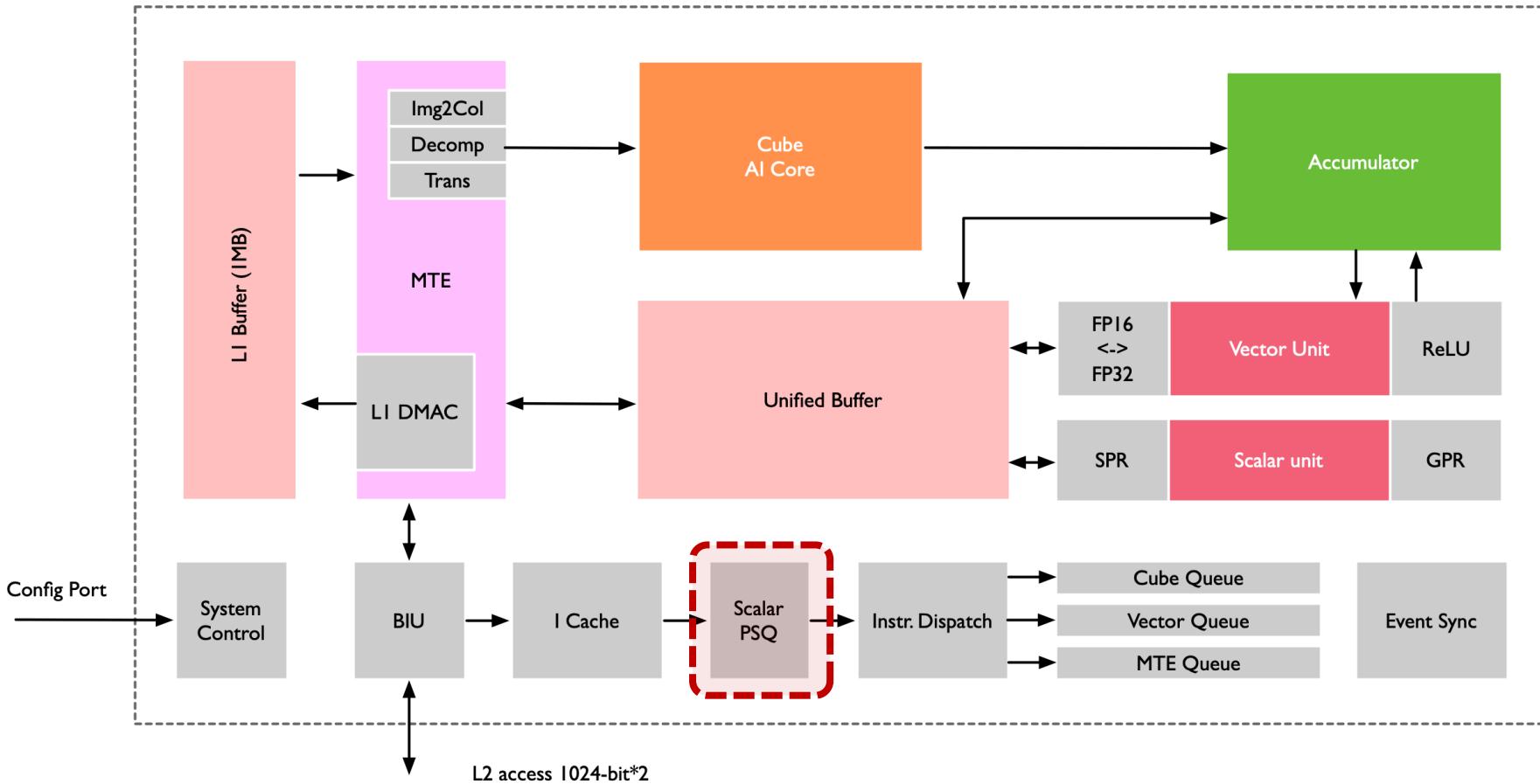
AI Core 存储单元：存储控制单元、缓冲区和寄存器组成



指令缓存：

- 在指令执行过程中，可以提前预取后续指令，并一次读入多条指令进入缓存，提升指令执行效率；

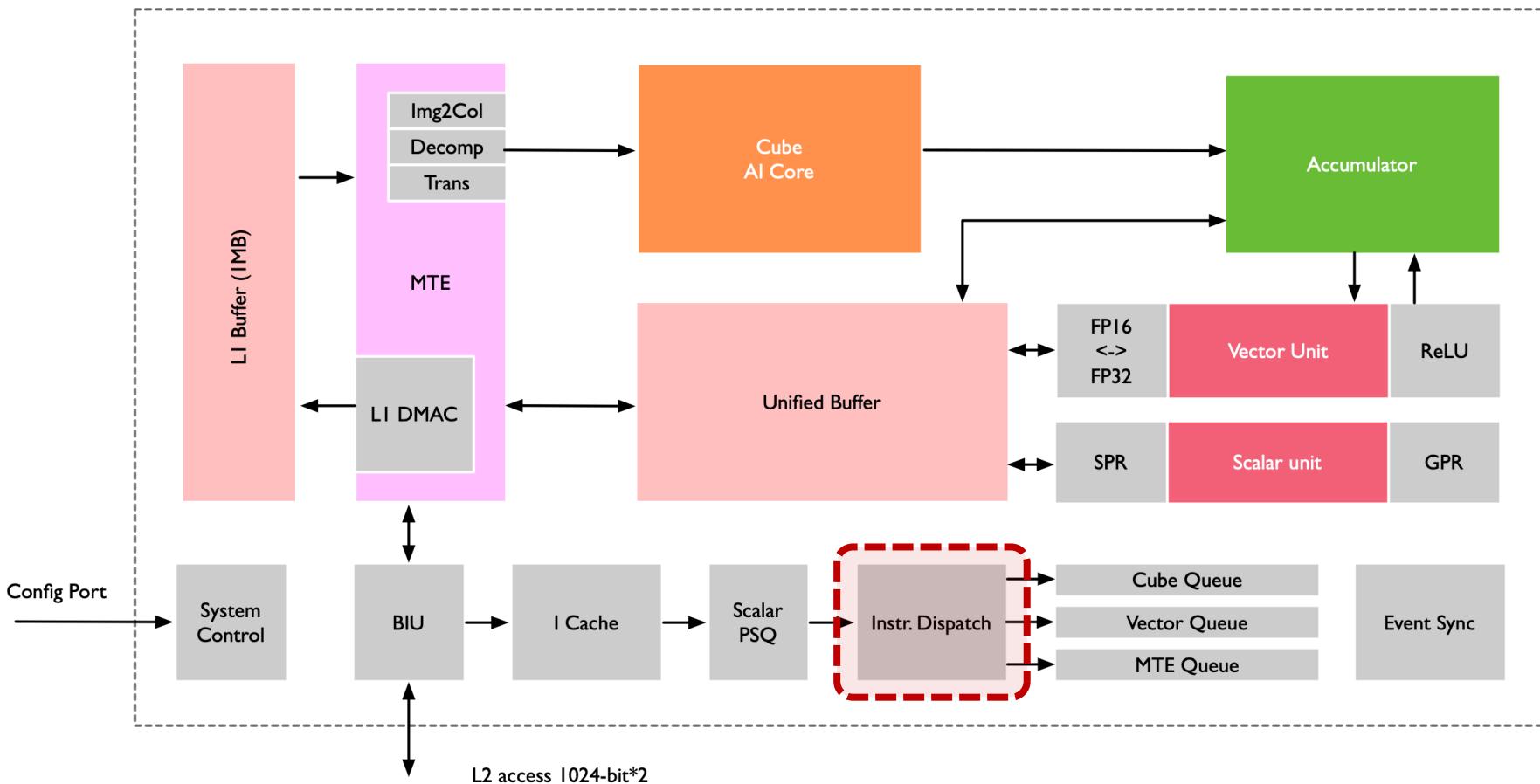
AI Core 存储单元：存储控制单元、缓冲区和寄存器组成



标量指令处理队列：

- 指令被解码后便会被导入标量队列中，实现地址解码与运算控制，这些指令包括矩阵计算指令、向量计算指令以及存储转换指令等；

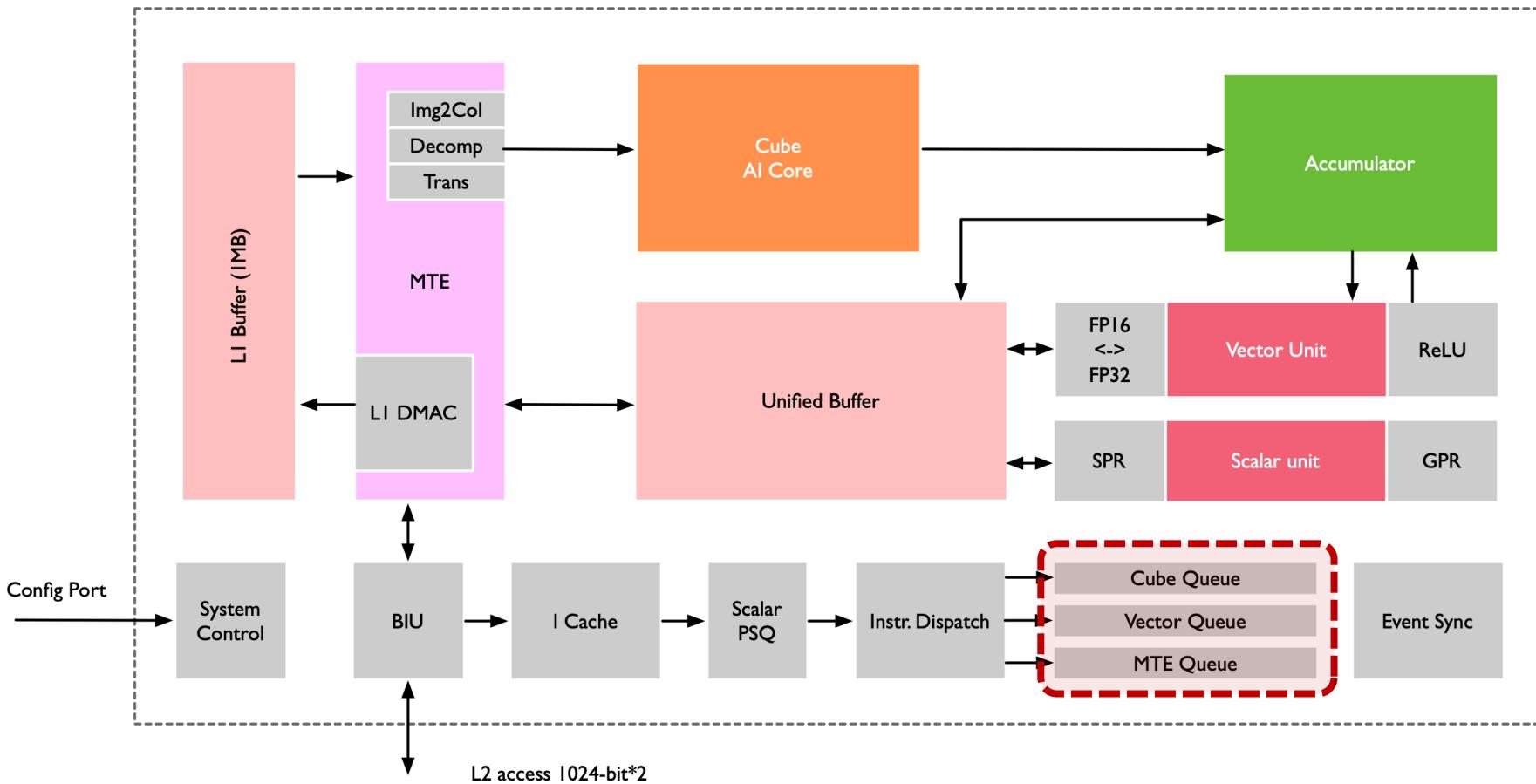
AI Core 存储单元：存储控制单元、缓冲区和寄存器组成



指令发射模块：

- 读取标量指令队列中配置好的指令地址和参数解码，然后根据指令类型分别发送到对应的指令执行队列中，而标量指令会驻留在标量指令处理队列中进行后续执行；

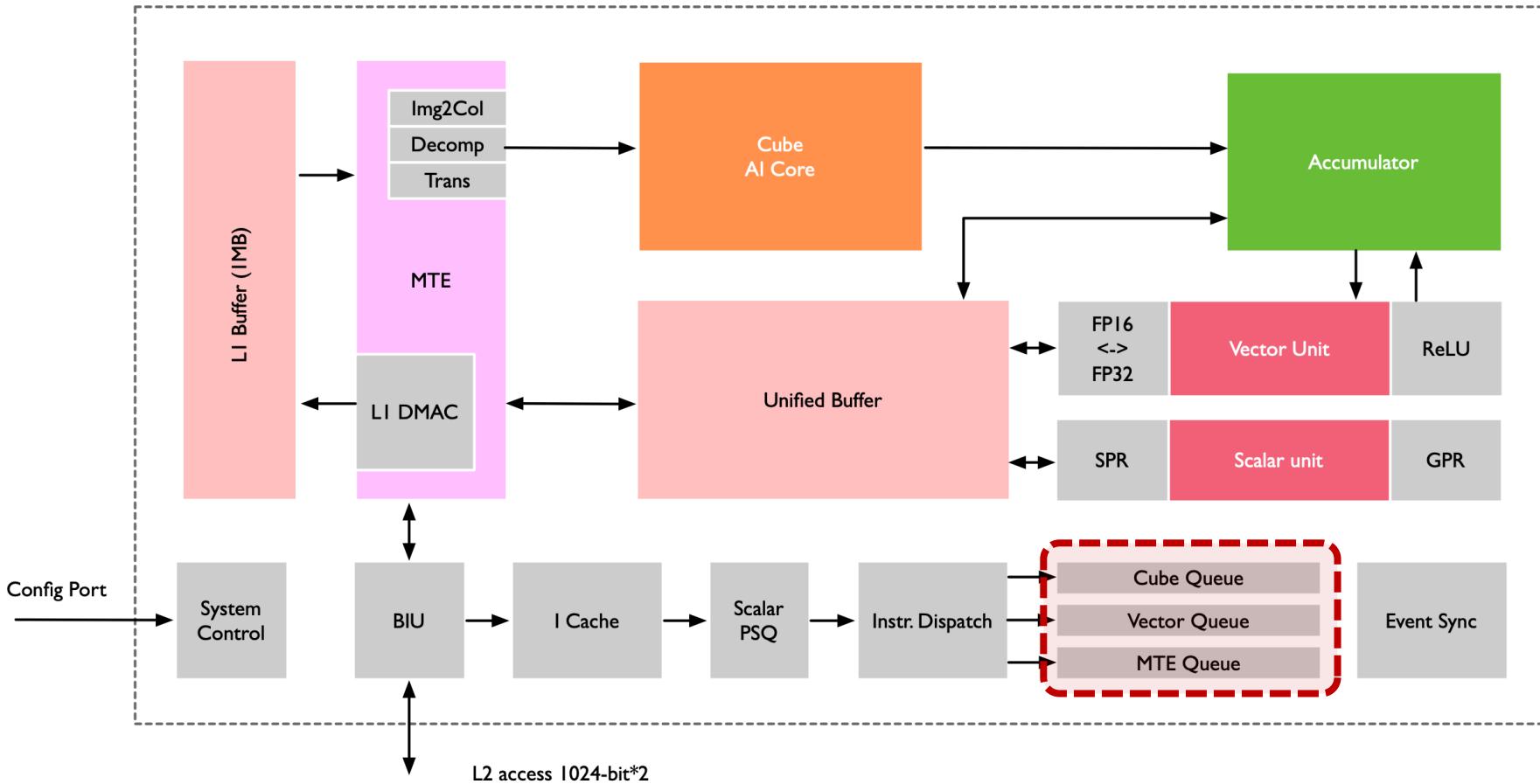
AI Core 存储单元：存储控制单元、缓冲区和寄存器组成



指令执行队列：

- 指令执行队列由矩阵运算队列、向量运算队列和存储转换队列组成，不同的指令进入相应的运算队列，队列中的指令按进入顺序执行；

AI Core 存储单元：存储控制单元、缓冲区和寄存器组成



事件同步模块：

- 时刻控制每条指令流水线的执行状态，并分析不同流水线的依赖关系，从而解决指令流水线之间的数据依赖和同步的问题。

04 小结与思考

思考

1. AICore 是专门针对矩阵乘 MAC 进行运算，跟英伟达的 Tensor Core 区别在哪里？
2. 了解完 NPU AICore 和 TPU 脉动阵列，你觉得 NPU/TPU 跟 GPU 的本质区别在哪里？





把AI系统带入每个开发者、每个家庭、
每个组织，构建万物互联的智能世界

Bring AI System to every person, home and
organization for a fully connected,
intelligent world.

Copyright © 2023 XXX Technologies Co., Ltd.
All Rights Reserved.

The information in this document may contain predictive statements including, without limitation, statements regarding the future financial and operating results, future product portfolio, new technology, etc. There are a number of factors that could cause actual results and developments to differ materially from those expressed or implied in the predictive statements. Therefore, such information is provided for reference purpose only and constitutes neither an offer nor an acceptance. XXX may change the information at any time without notice.



Course chenzomi12.github.io

GitHub github.com/chenzomi12/DeepLearningSystem