GPGPU L1 Data Cache coreReq/coreRsp spec

金楚丰

THU DSPLAB internal-use

coreReq Interface overview

```
class DCacheCoreReq
        instrId = UInt(WIdBits.W)
        isWrite = Bool()
        tag = UInt(TagBits.W)
        setIdx = UInt(SetIdxBits.W)
        perLaneAddr = Vec(NLanes, new DCachePerLaneAddr)
        data = Vec(NLanes, UInt(WordLength.W))

class DCachePerLaneAddr
        activeMask = Bool()
        blockOffset = UInt(BlockOffsetBits.W)
        wordOffset1H = UInt(BytesOfWord.W)
```

coreReq Interface Field Definition

- instrId
  - 由LSU生成，要求不同warp的不同指令有唯一编号。
  - 这一标识字段作用有二
    - 1、LSU用来聚合返回的uncoalesced请求
    - 2、L2返回时在TileLink中区分请求
- isWrite
  - 指令本体的一部分，区分是READ还是WRITE
- tag， setIdx
  - 所有Lane共用，地址字段的一部分
- activeMask
  - 每个Lane独有，来自Mask寄存器，指示当前Lane是否活跃
- blockOffset
  - 每个Lane独有，由地址字段和访存类型转换而来
- wordOffset1H
  - 每个Lane独有，由地址字段和访存类型转换而来，

THU DSPLAB internal-use

tag setIdx blockOffset Definition

- tag， setIdx
  - 地址字段的一部分
  - 所有Lane共用
  - 一般是最高两个字段
- blockOffset
  - 每个Lane独有
  - 介于setIdx和wordOffset之间的字段

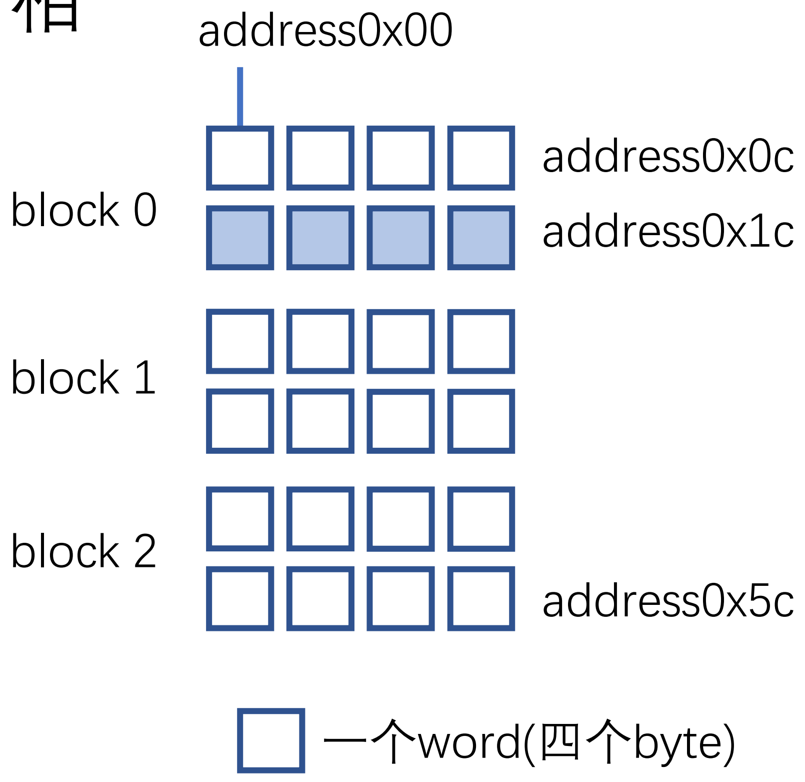
```
//                                                 |   blockOffset  |
//                                               bankOffset      wordOffset
// |32      tag       22|21   setIdx   11|10 9|8 bankIdx 2|1 0|
```

blockOffset Cases

- Nlane = 4, BlockSize = 8
- tag， setIdx， wordOffset1H与blockOffset四者相互独立

- A coalesced request of successive address
  - VL* unit-stride
  - base addr(LSB 8 bits) = 0x10

	Lane1	Lane2	Lane3	Lane4
blockOffset (3 bits)	b100	b101	b110	b111
activeMask	1	1	1	1

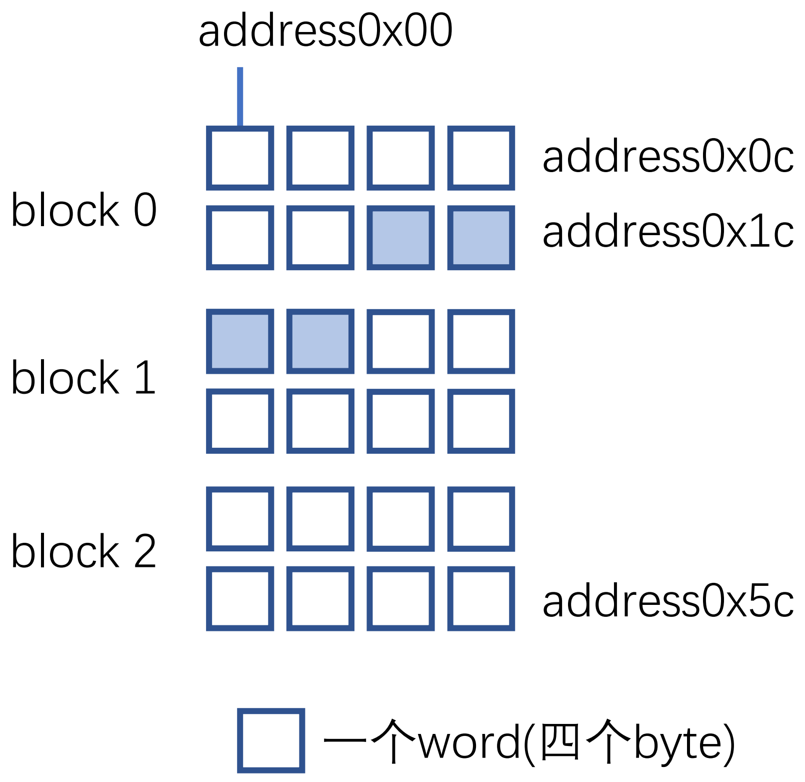


blockOffset Cases

- An uncoalesced request of successive address
  - VL unit-stride
  - base addr(LSB 8 bits) = 0x18

Req in cycle 1	Lane1	Lane2	Lane3	Lane4
blockOffset (3 bits)	b110	b111	x	x
activeMask	1	1	0	0

Req in cycle 2	Lane1	Lane2	Lane3	Lane4
blockOffset (3 bits)	x	x	b000	b001
activeMask	0	0	1	1

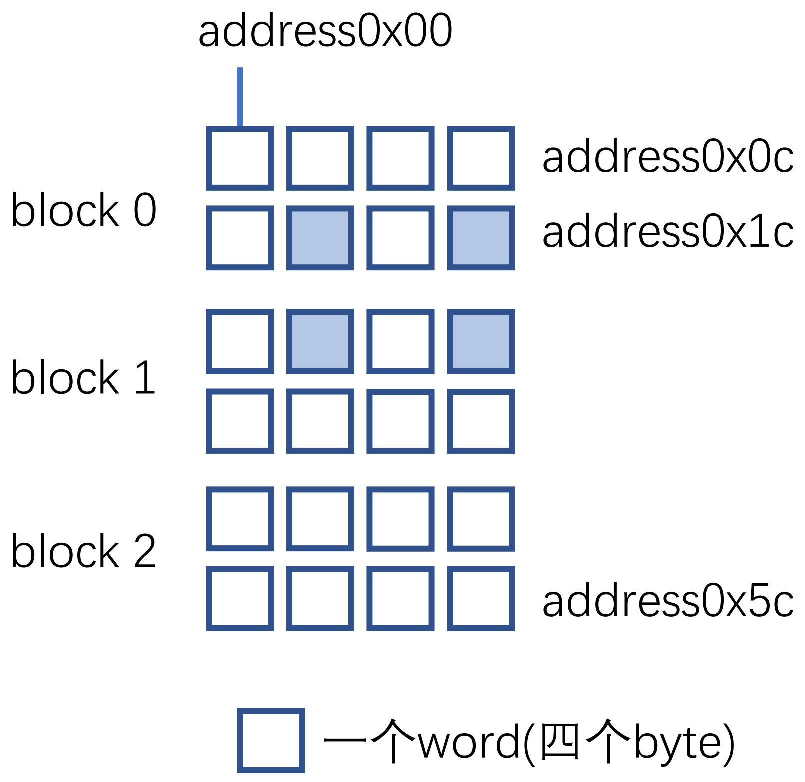


blockOffset Cases

- An uncoalesced request of jump address
  - VLS strided
  - stride = 2
  - base addr(LSB 8 bits) = 0x14

Req in cycle 1	Lane1	Lane2	Lane3	Lane4
blockOffset (3 bits)	b101	b111	x	x
activeMask	1	1	0	0

Req in cycle 2	Lane1	Lane2	Lane3	Lane4
blockOffset (3 bits)	x	x	b001	b011
activeMask	0	0	1	1

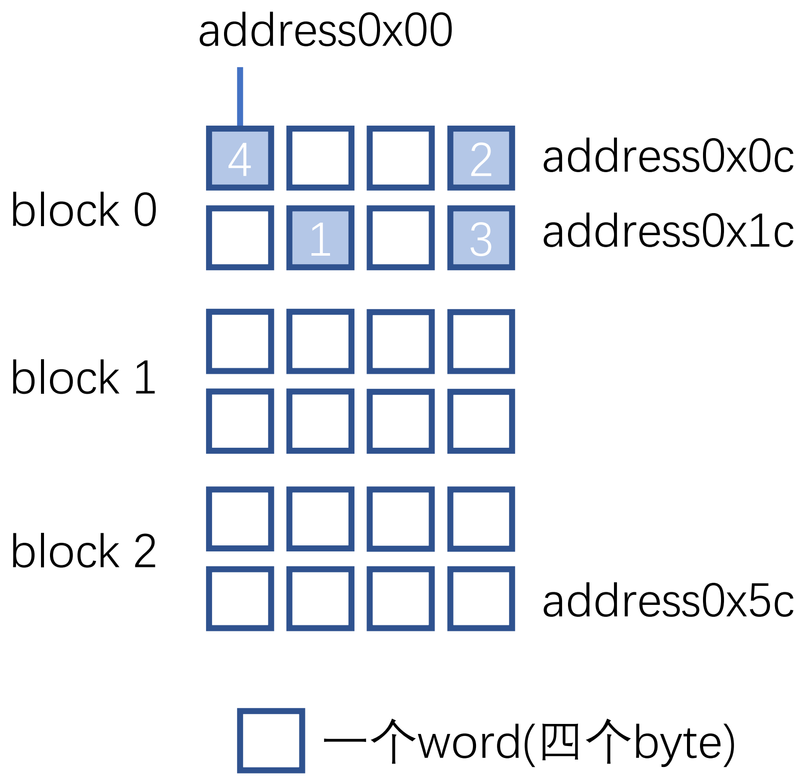


blockOffset Cases

- A coalesced request of scatter-gather
  - VLX indexed
  - 注意core Lane的请求顺序和内存数据的顺序可以不同

Req in cycle 1	Lane1	Lane2	Lane3	Lane4
blockOffset (3 bits)	b101	b011	b111	b000
activeMask	1	1	1	1

- 这个例子挖掘了core-mem乱序重排的最大带宽



blockOffset Cases

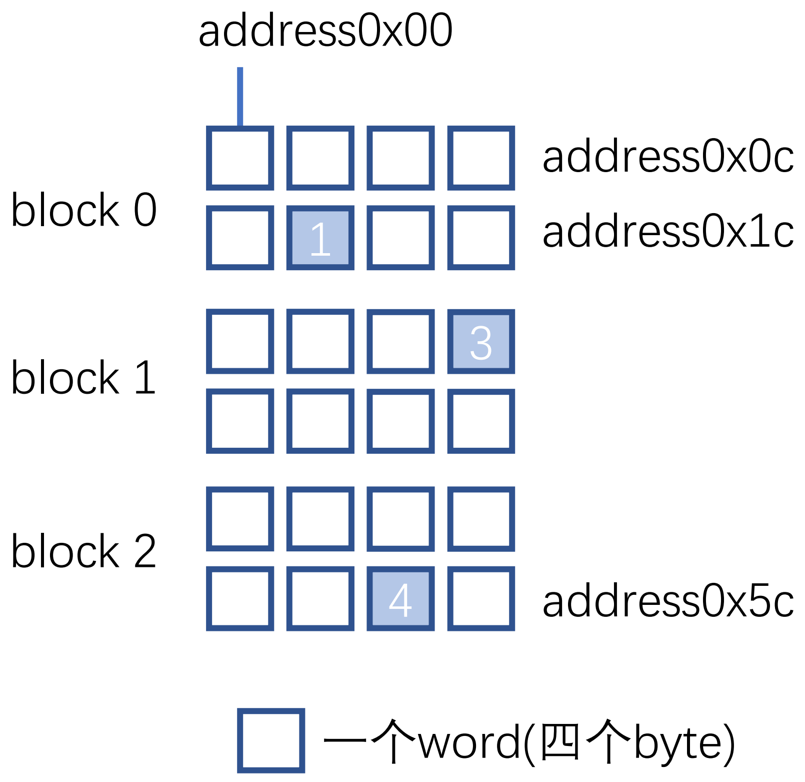
- An uncoalesced request of scatter-gather
  - VLX indexed
  - 本例中4个Lane被mask off一个

Req in cycle 1	Lane1	Lane2	Lane3	Lane4
blockOffset (3 bits)	b101	x	x	x
activeMask	1	0	0	0

Req in cycle 2	Lane1	Lane2	Lane3	Lane4
blockOffset (3 bits)	x	x	b011	x
activeMask	0	0	1	0

Req in cycle 3	Lane1	Lane2	Lane3	Lane4
blockOffset (3 bits)	x	x	x	b110
activeMask	0	0	0	1

THU DSPLAB internal-use



wordOffset1H Definition

- wordOffset1H
  - 每个Lane独有
  - 指示当前访问是Word，Half word还是Byte
  - 原始地址字段中的最低2 bit，但这里需要扩展为4 bit：
- 举例如下：

case 1 addrLSB = ...100, byte access
wordOffset1H = 0001
case 2 addrLSB = ...110, byte access
wordOffset1H = 0100
case 3 addrLSB = ...100, half word access
wordOffset1H = 0011
case 4 addrLSB = ...101, half word access
illegal
case 5 addrLSB = ...110, half word access
wordOffset1H = 1100

case 6 addrLSB = ...100, word access
wordOffset1H = 1111
case 789 addrLSB = ...101, word access
addrLSB = ...110, word access
addrLSB = ...111, word access
illegal

THU DSPLAB internal-use

coreRsp Interface

```
class DCacheCoreRsp
        instrId = UInt(WIdBits.W)
        data = Vec(NLanes, UInt(WordLength.W))
        activeMask = Vec(NLanes, Bool())
```

- 可想而知，每次返回的数据肯定都位于同一个block