NandSigZoneInfo

Nand中的page0

```
+lifetime: unsigned int
+badblock: unsigned int
+ZoneID: unsigned short
```

NandZoneInfo

Nand中Page1

```
+validPage: int
+lifetime: unsigned int
+serialnumber: unsigned int
+badblock: unsigned int
+preZone: ZoneInfo
+nextZone: ZoneInfo
+ZoneID: int
+MagicID
```

NandPageInfo

Nand中的Page3

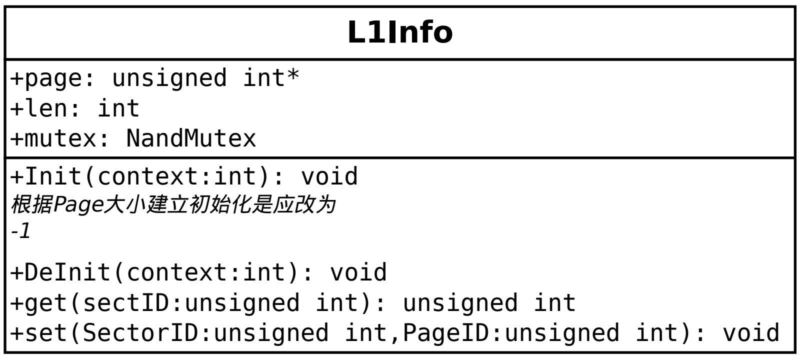
```
+NextPageInfo: unsigned short
+ZoneID: unsigned short
+CompressLen: unsigned short
+CompressData[]: unsigned char
+MagicID: unsigned short
+crc: unsigned short
+NextCompresspageID: unsigned short
```

Hash

在这里的功能不提供线程保护，需要使
用者去保护。

```
+top: HashNode**
+usezone_count: unsigned int
+maxlifetime: unsigned int
+minlifetime: unsigned int
+zoneID_count: int
+first_pos: int
+prev_pos: int
+find_lifetime: unsigned int
```

```
+init(hash:Hash**,top:SigZoneInfo*,zoneid:unsigned short*,
      count:int): int
+deinit(hash:Hash**): void
+Insert(hash:Hash*,szi:SigZoneInfo*): int
+delete(hash:Hash*,szi:SigZoneInfo*): int
+FindFirstLessLifeTime(hash:Hash*,lifetime:unsigned int,
                       szi:SigZoneInfo**): int
+FindNextLessLifeTime(hash:Hash*,prev:int,
                      szi:SigZoneInfo**): int
+getminlifetime(hash:Hash*): unsigned int
+getmaxlifetime(hash:Hash*): unsigned int
+getcount(hash:Hash*): unsigned int
```

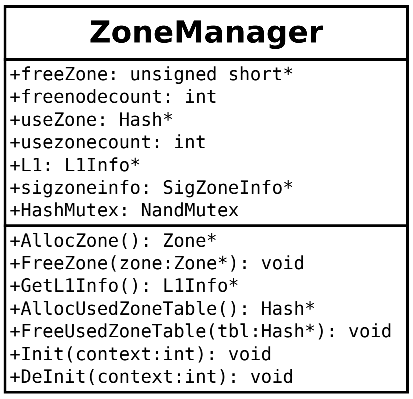


L1Info

```
+page: unsigned int*
+len: int
+mutex: NandMutex
```

```
+Init(context:int): void
```
根据Page大小建立初始化是应改为
-1

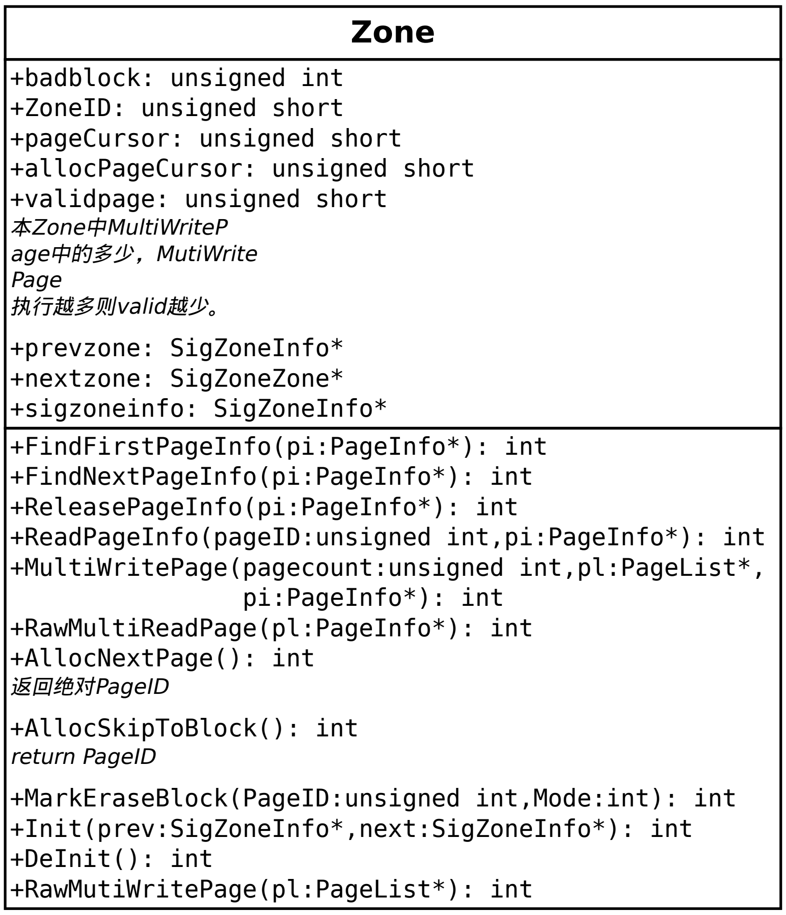
```
+DeInit(context:int): void
+get(sectID:unsigned int): unsigned int
+set(SectorID:unsigned int,PageID:unsigned int): void
```



ZoneManager

```
+freeZone: unsigned short*
+freenodecount: int
+useZone: Hash*
+usezonecount: int
+L1: L1Info*
+sigzoneinfo: SigZoneInfo*
+HashMutex: NandMutex
```

```
+AllocZone(): Zone*
+FreeZone(zone:Zone*): void
+GetL1Info(): L1Info*
+AllocUsedZoneTable(): Hash*
+FreeUsedZoneTable(tbl:Hash*): void
+Init(context:int): void
+DeInit(context:int): void
```



Zone

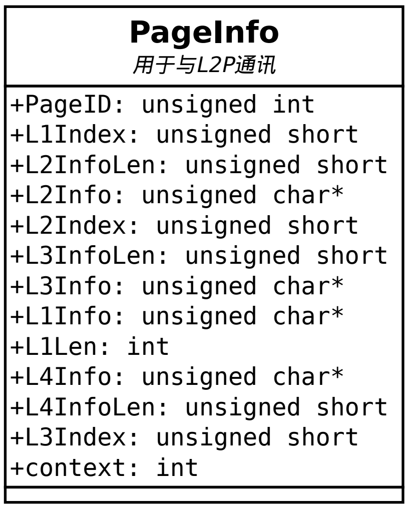
```
+badblock: unsigned int
+ZoneID: unsigned short
+pageCursor: unsigned short
+allocPageCursor: unsigned short
+validpage: unsigned short
```
本Zone中MultiWriteP
age中的多少，MutiWrite
Page
执行越多则valid越少。

```
+prevzone: SigZoneInfo*
+nextzone: SigZoneZone*
+sigzoneinfo: SigZoneInfo*
```

```
+FindFirstPageInfo(pi:PageInfo*): int
+FindNextPageInfo(pi:PageInfo*): int
+ReleasePageInfo(pi:PageInfo*): int
+ReadPageInfo(pageID:unsigned int,pi:PageInfo*): int
+MultiWritePage(pagecount:unsigned int,pl:PageList*,
                pi:PageInfo*): int
+RawMultiReadPage(pl:PageInfo*): int
+AllocNextPage(): int
```
返回绝对PageID

```
+AllocSkipToBlock(): int
```
return PageID

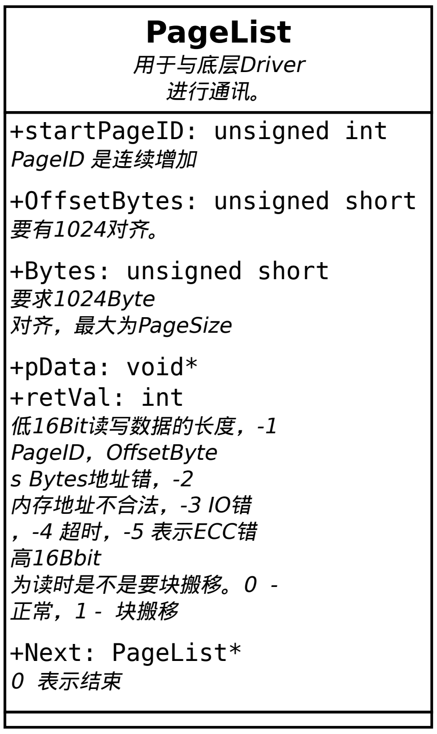
```
+MarkEraseBlock(PageID:unsigned int,Mode:int): int
+Init(prev:SigZoneInfo*,next:SigZoneInfo*): int
+DeInit(): int
+RawMutiWritePage(pl:PageList*): int
```



PageInfo

用于与L2P通讯

```
+PageID: unsigned int
+L1Index: unsigned short
+L2InfoLen: unsigned short
+L2Info: unsigned char*
+L2Index: unsigned short
+L3InfoLen: unsigned short
+L3Info: unsigned char*
+L1Info: unsigned char*
+L1Len: int
+L4Info: unsigned char*
+L4InfoLen: unsigned short
+L3Index: unsigned short
+context: int
```



PageList

*用于与底层Driver
进行通讯。*

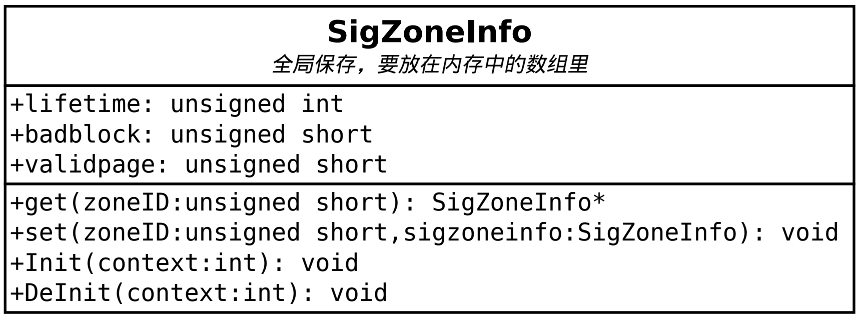
+startPageID: unsigned int
PageID 是连续增加

+OffsetBytes: unsigned short
要有1024对齐。

+Bytes: unsigned short
*要求1024Byte
对齐，最大为PageSize*

+pData: void*
+retVal: int
*低16Bit读写数据的长度，-1
PageID，OffsetByte
s Bytes地址错，-2
内存地址不合法，-3 IO错
，-4 超时，-5 表示ECC错
高16Bbit
为读时是不是要块搬移。0 -
正常，1 - 块搬移*

+Next: PageList*
0 表示结束

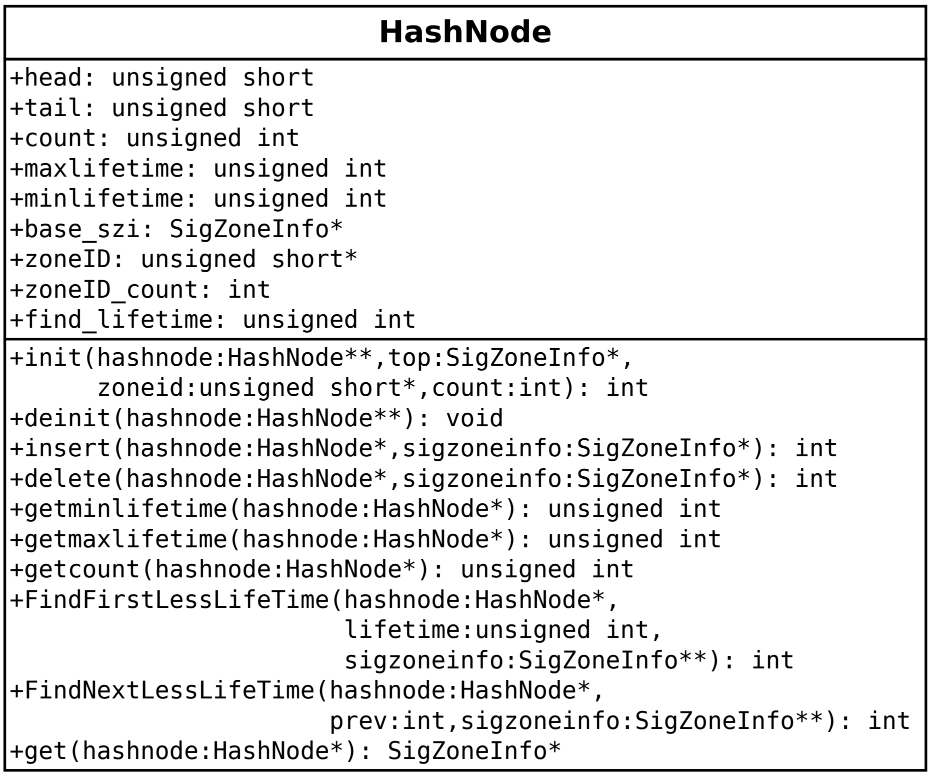


SigZoneInfo

全局保存，要放在内存中的数组里

```
+lifetime: unsigned int
+badblock: unsigned short
+validpage: unsigned short
```

```
+get(zoneID:unsigned short): SigZoneInfo*
+set(zoneID:unsigned short,sigzoneinfo:SigZoneInfo): void
+Init(context:int): void
+DeInit(context:int): void
```



```
HashNode
+head: unsigned short
+tail: unsigned short
+count: unsigned int
+maxlifetime: unsigned int
+minlifetime: unsigned int
+base_szi: SigZoneInfo*
+zoneID: unsigned short*
+zoneID_count: int
+find_lifetime: unsigned int
+init(hashnode:HashNode**,top:SigZoneInfo*,
      zoneid:unsigned short*,count:int): int
+deinit(hashnode:HashNode**): void
+insert(hashnode:HashNode*,sigzoneinfo:SigZoneInfo*): int
+delete(hashnode:HashNode*,sigzoneinfo:SigZoneInfo*): int
+getminlifetime(hashnode:HashNode*): unsigned int
+getmaxlifetime(hashnode:HashNode*): unsigned int
+getcount(hashnode:HashNode*): unsigned int
+FindFirstLessLifeTime(hashnode:HashNode*,
                       lifetime:unsigned int,
                       sigzoneinfo:SigZoneInfo**): int
+FindNextLessLifeTime(hashnode:HashNode*,
                      prev:int,sigzoneinfo:SigZoneInfo**): int
+get(hashnode:HashNode*): SigZoneInfo*
```