

# 总线（下）

内存每次寻址的延迟很大，但一次寻址得到的数据块通常远大于总线宽度。

引入突发传输的机制，在一次访存事务中传输连续的多个字。

缓存总线（Cache Bus, CBus）借鉴了 AXI4，用于支持突发传输。

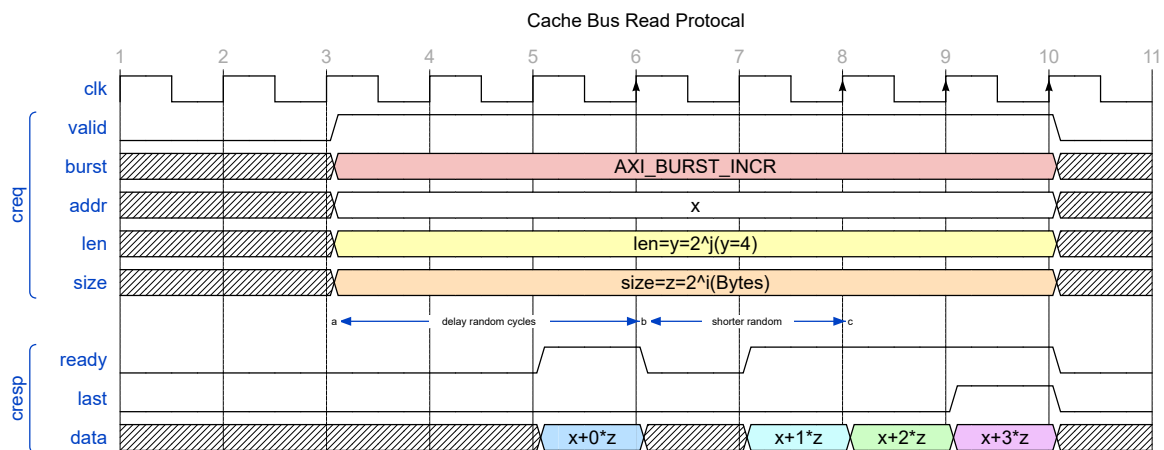
```
1  typedef struct packed {
2      logic    valid;    // in request?
3      logic    is_write; // is it a write transaction?
4      msize_t  size;     // number of bytes in one burst
5      addr_t   addr;     // start address
6      strobe_t strobe;   // which bytes are enabled?
7      word_t   data;     // the data to write
8      mlen_t   len;      // number of bursts
9      axi_burst_type_t burst;
10 } cbus_req_t;
11
12 typedef struct packed {
13     logic ready; // is data arrived in this cycle?
14     logic last;  // is it the last word?
15     word_t data; // the data from AXI bus
16 } cbus_resp_t;
```

CBus 有三个用于突发传输的请求信号：

- size 表示一个字的大小，需要小于等于总线宽度
- len 表示这次突发传输中字的数量
- burst 表示突发传输的模式，比如 AXI\_BURST\_FIXED，AXI\_BURST\_INCR
  - AXI\_BURST\_FIXED 表示不使用突发传输，len 也为 MLEN1
  - AXI\_BURST\_INCR 表示使用递增式突发传输，传输 `addr`，`addr + size`，`addr + (len - 1) * size` 位置的字。

CBus 中有两个握手信号 `ready` 和 `last`，分别表示一个字和整个事务的握手。

读请求：



写请求:

