验证方案：

|  |  |  |  |
| --- | --- | --- | --- |
| 测试功能点 | 测试内容 | 测试通过标准 | 测试类名 |
| 读数据测试 | 发送读请求，读地址为随机值，检查读出的数据是否正确。对写数据和写字节选通信号进行随机化，检查是否会对DUT的行为产生影响。 | 每次读出的值都正确，并且覆盖率达到100% | read\_test |
| 写数据测试 | 1. 首先发送读请求，把cache全部填满，再发送命中的写请求，写数据指定为写地址，写字节选通信号为{8’b1, 8’b11, 8’b1111, 8’hff}中的随机值。完成写操作后再对相同的地址进行读，检查读出的数据是否为写入后的值。  然后再发送读操作，将cache中原有的数据替换出去，然后再发送读操作，将刚刚替换出去的数据再读回cache中，已验证脏数据的写回是否成功。  2. 发送写请求，把cache填满，写数据指定为写地址，写字节选通信号为{8’b1, 8’b11, 8’b1111, 8’hff}中的随机值。完成写操作后再对相同的地址进行读，检查读出的数据是否为写入后的值。 | 每次读出的值都正确，并且wstrb的覆盖率达到100% | write\_test |
| 替换算法测试 | 先将cache中的两组cache line全部填满，再对第一组进行读或者写操作（此时第一组数据为最新被访问的数据，应该不会被替换），然后访问一组新的数据，然后再次访问第一组cache line中的数据，检查第一组数据是否命中。 | 能否在指定的时钟周期返回正确的数据。 | replace\_algorithm\_test |
| 地址对齐访问测试 | 发送读写请求，读写地址为非64位对其的地址，查看DUT是否能够忽略地址的低三位。 | 读的值是否正确 | address\_align\_test |
| 随机读写测试 | 发送随机的读写请求，检查行为是否与对memory直接进行读写的行为保持一致。 | 读的值是否正确 | random\_read\_write\_test |