A REPORT ON

Cache Memory Controller Design

BY

SUKRUTH S

2020H1400236H

M.E. EMBEDDED SYSTEMS

Prepared in fulfilment of the

(MEL G624)

Advanced VLSI Architecture

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI

(NOVEMBER 2021)

ABSTRACT

The controller cache is a physical memory region that speeds up two types of I/O (input/output) operations: communication between controllers and hosts, and communication between controllers and discs. The hosts and controllers interact via high-speed links to read and write data. However, because discs are sluggish devices, communications from the controller's back end to the discs are slower.

When the controller cache gets data, it notifies the host apps that the data is now in its possession. The host programmes will not have to wait for the I/O to be written to disc in this manner. Instead, apps can carry on with their work. Server programmes can also access the cached data directly, reducing the need for further disc accesses. In this project report we have implemented a simple cache controller and its simulation is done.

Contents

List of figures

Block-level Schematic Design

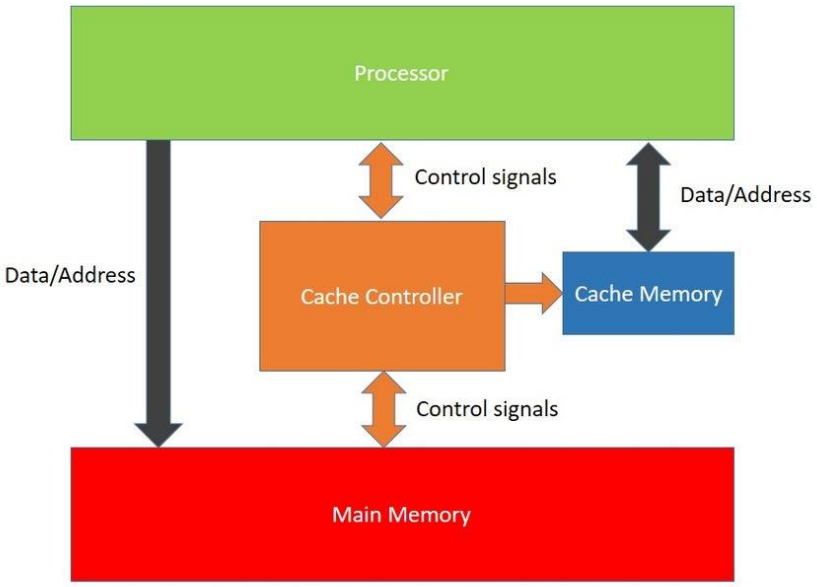


Figure 1. Block level diagram of top module

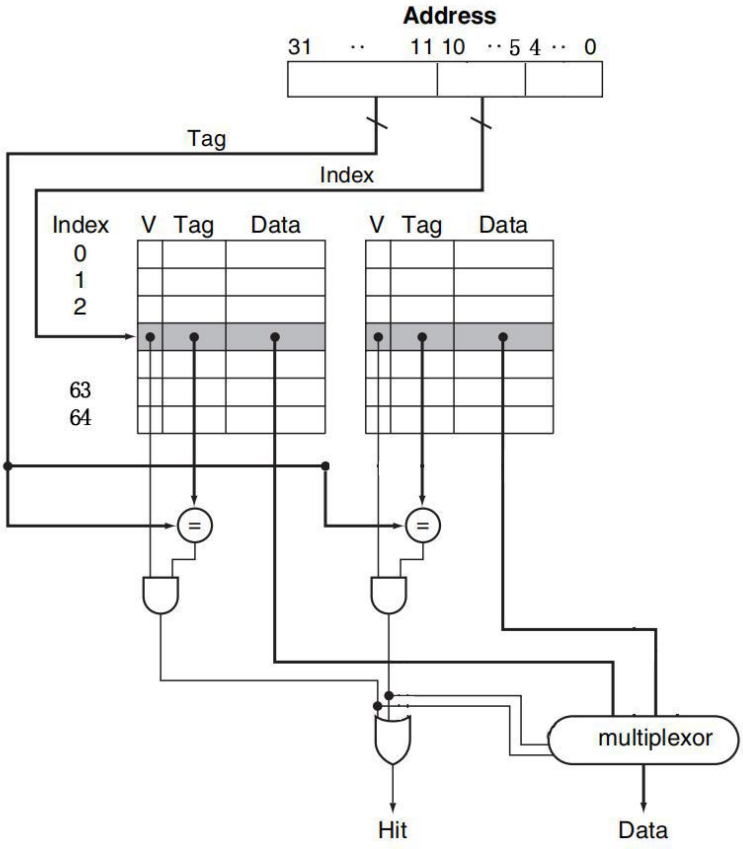


Figure 2.Cache block working

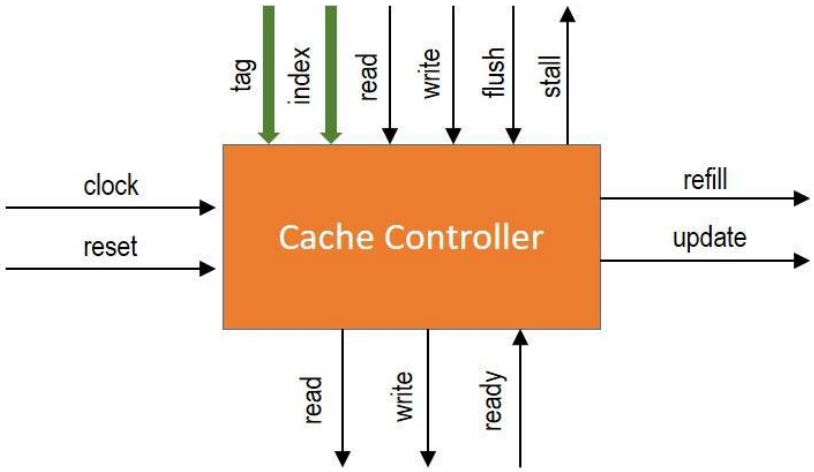


Figure 3. Block level representation of cache controller

Specification

- Cache controller with direct mapping.
- Specifically designed for single-banked caches.
- On write hits, there is a write-through policy.
- 'No-Write-Allocate' or 'No-Write-Allocate' On write misses, there is a Write-Around policy.
- There is a Tag Array in there.
- Index and Tag widths can be customised.
- There is not a Write Buffer or any other optimization.

Design

- 128 words can be stored in main memory.
- The cache memory may hold up to 16 words.
- The size of a block is four bytes.
- The CPU produces a 7-bit physical address, with the least significant two bits designated as "block offset" and the most significant five bits indicating the block number.

- Similarly, the cache address is separated into "tag bits" (the most significant three bits), "line numbers" (the next two bits), and "block offset" (the least significant two bits).

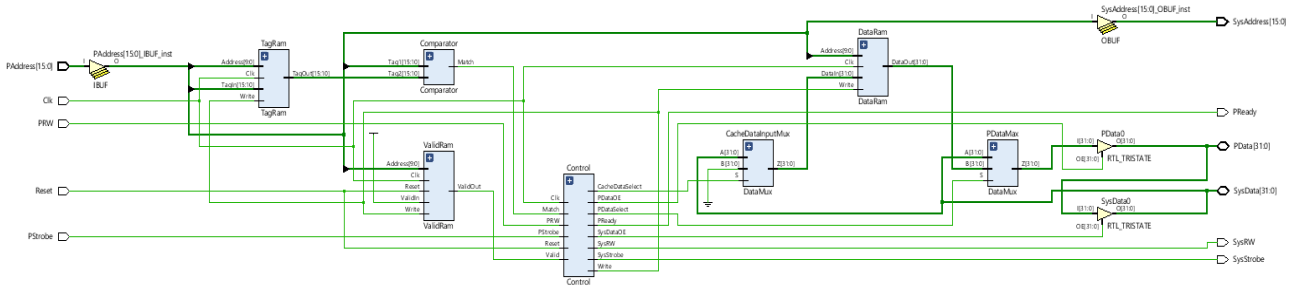


Figure 4.RTL of top module

FSM

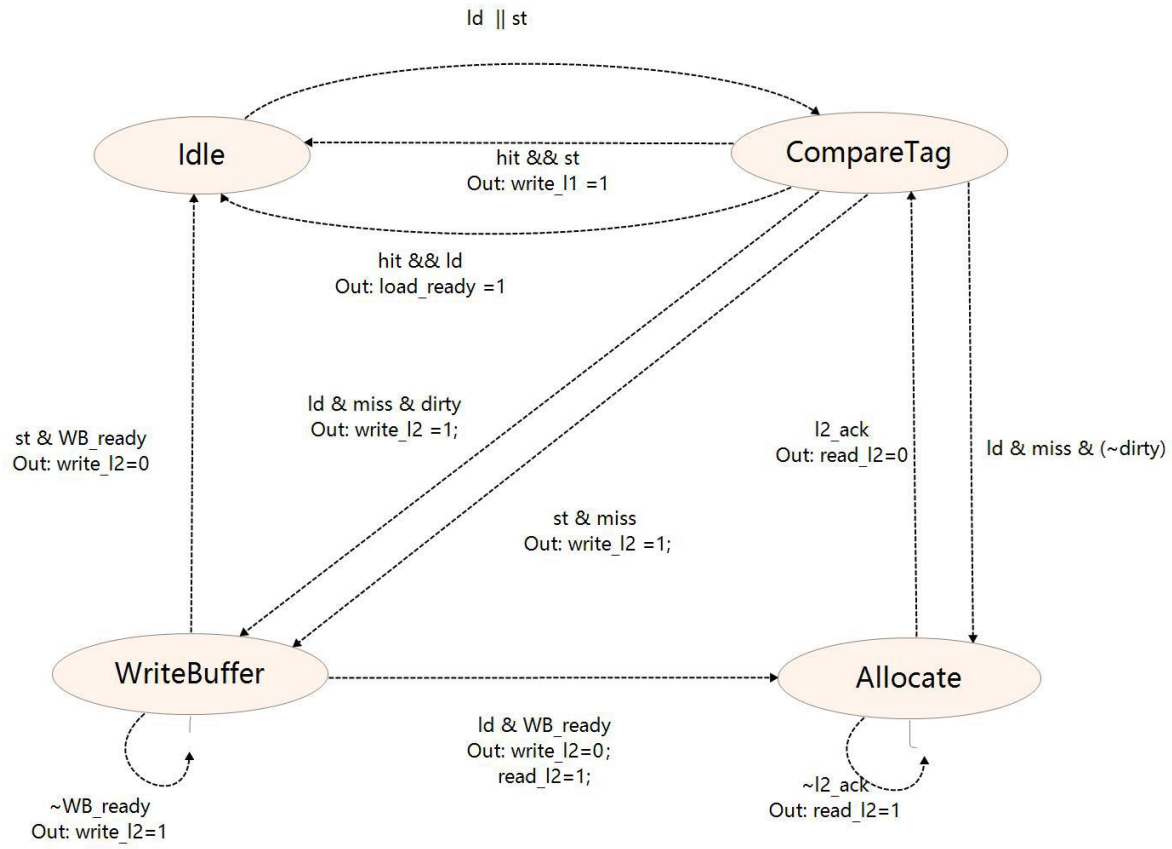


Figure 5.Cache FSM

- IDLE: The FSM remains in this state until it receives a valid read or write request from the processor, at which point it is moved to the Compare Tag state.
- This state checks to determine if the requested read or write is successful or not. The tag to be compared is chosen by the index section of the URL. It is a hit if the data in the cache block referenced by the index component of the address is correct and the tag portion of the address matches the tag. If it is a load, the data is read from the specified word; if it is a store, the data is written to the designated word.

- Write Allocate: When a write miss occurs, the CPU enters this state, and the data is written to both the cache and the main memory, updating the copies of the data in both locations.
- No Write Allocate: When a write request is received, the processor enters this state, where the data is exclusively written to the cache. As a result, the data copy in main memory would not be updated.

Induvial modules

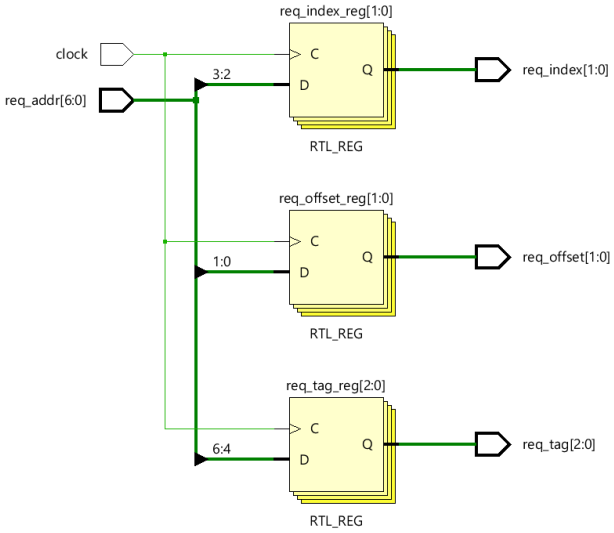


Figure 6.Address separator

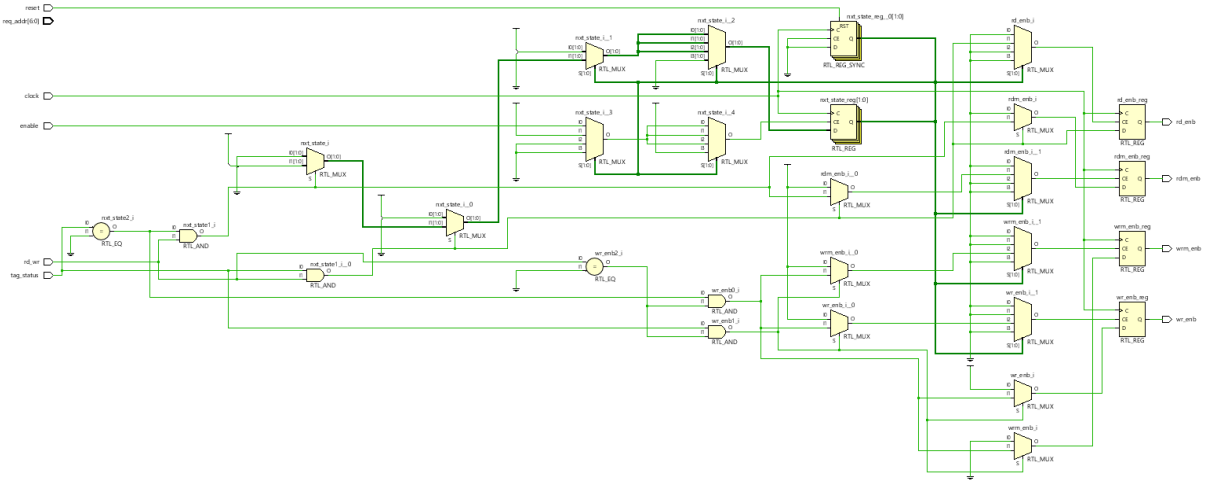


Figure 7.Cache controller

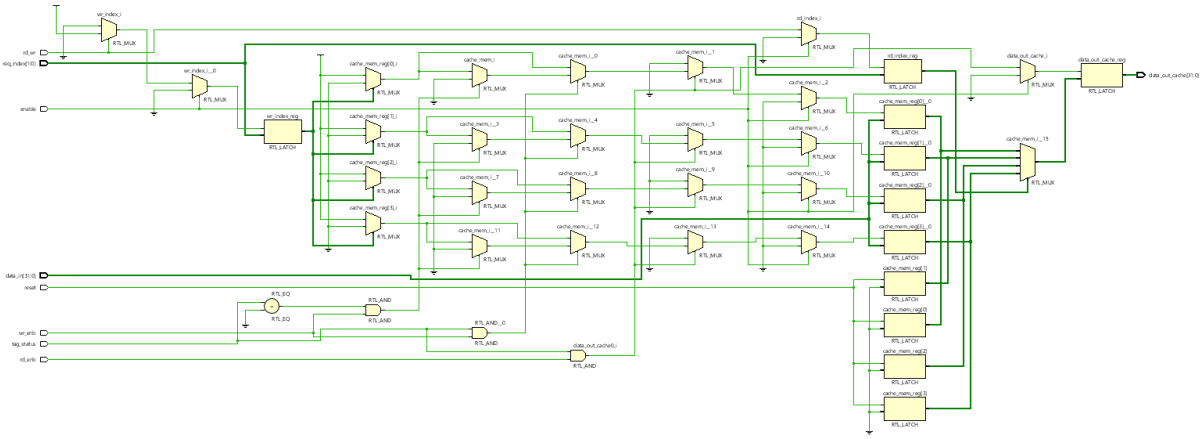


Figure 8.Cache memory

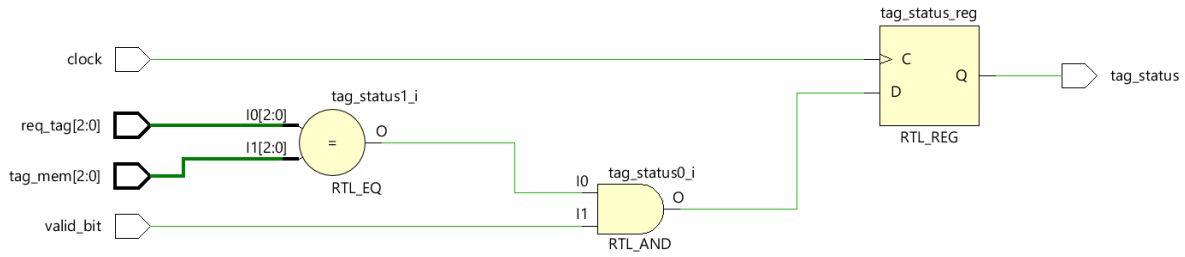


Figure 9.Tag comparator

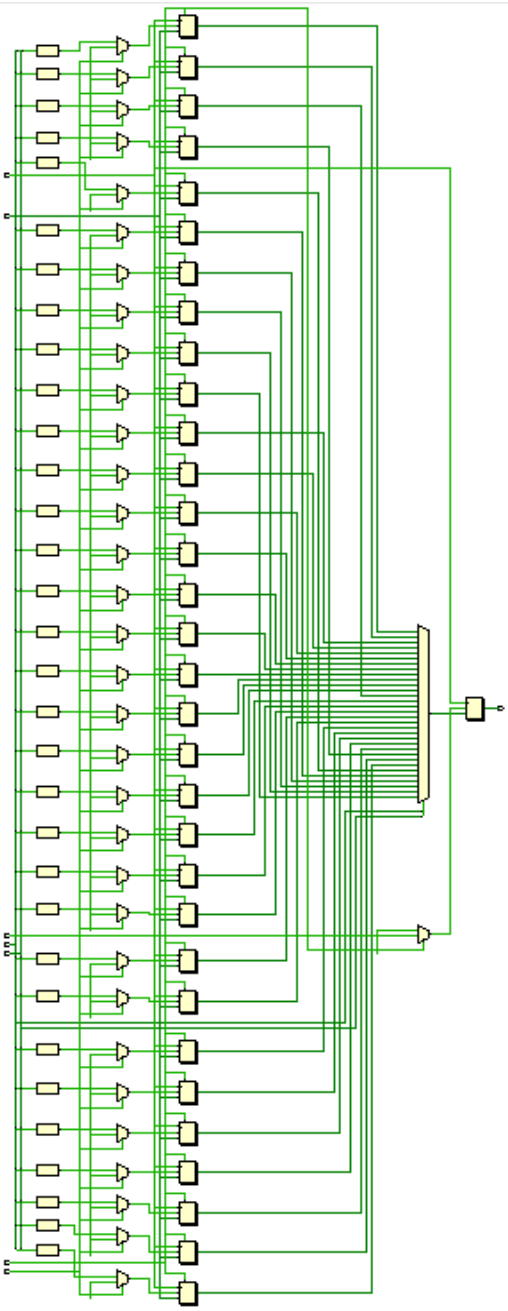


Figure 10.Main memory

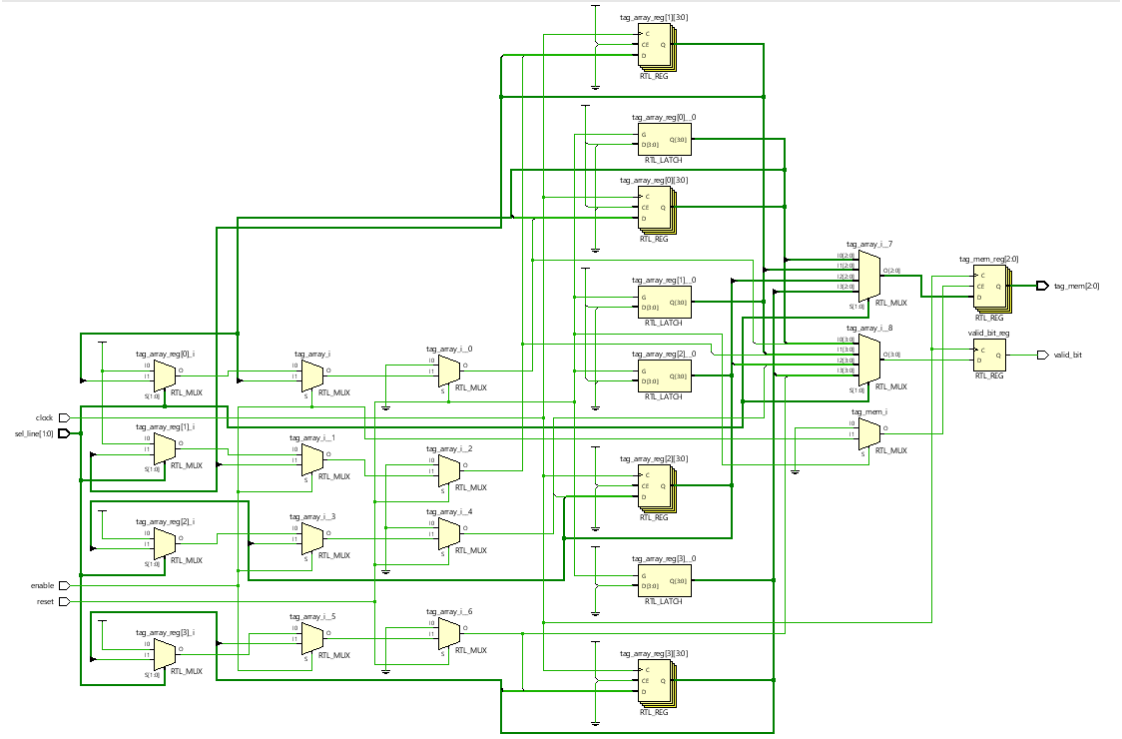


Figure 11.Tag cache

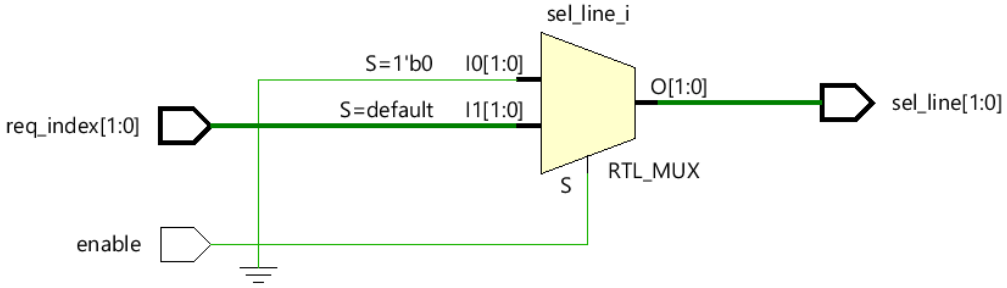


Figure 12.Decoder

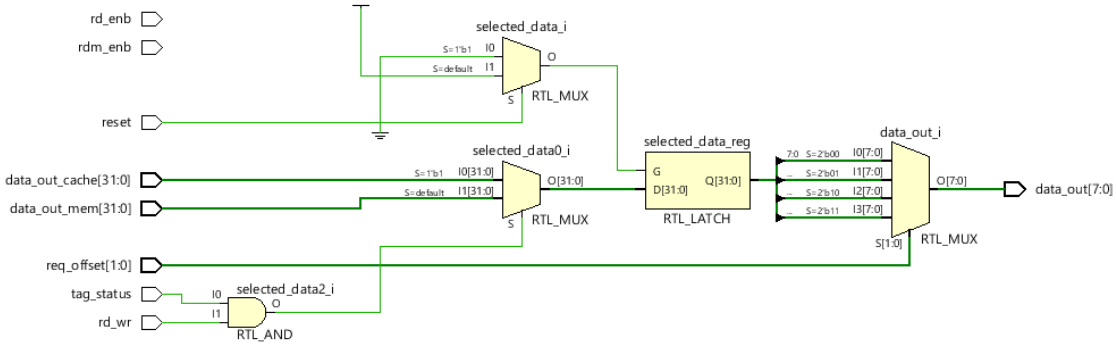


Figure 13.Mux

Simulation results

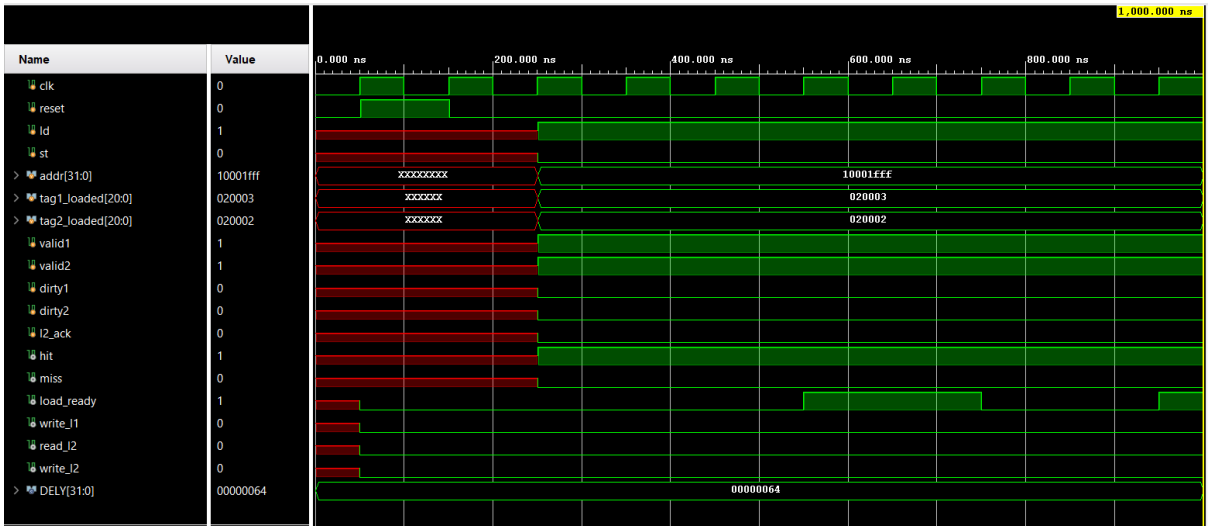


Figure 14.All test vectors results in simulation

See module cache controller tb for the test program.

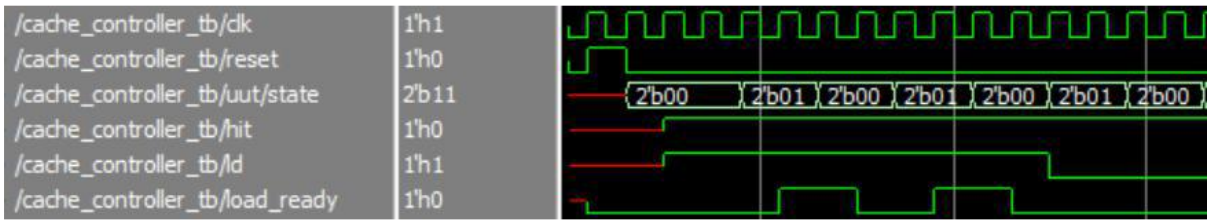
The controller is primarily interested in determining if the state's transition logic and signal output are proper. It may be altered by altering the valid bit of the data instead of changing the addr all the time to accomplish cache hit/miss, but for the purpose of simplicity, the utilised addr and two tag loaded data have been left intact, as shown in Figure 9, where tag1 loaded, etc.

tag2 loaded is not identical to addr[31:11] in addr[31:11].

Later on, these three signals will not be mentioned.

Induvial cases

- read hit

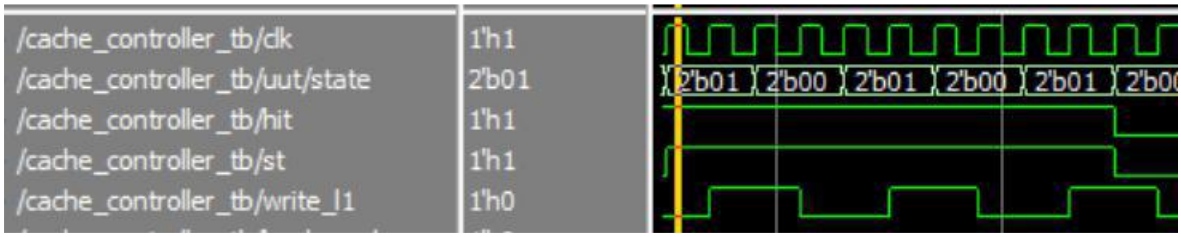


Explanation of each digital label:

- Prerequisite: The controller is initialized, reset=0, state = 2'b00 is idle-Idle state
- hi1=1, ld=1, read hit, state switch, from 00 (Idle) → 01 (CompareTag) → 00

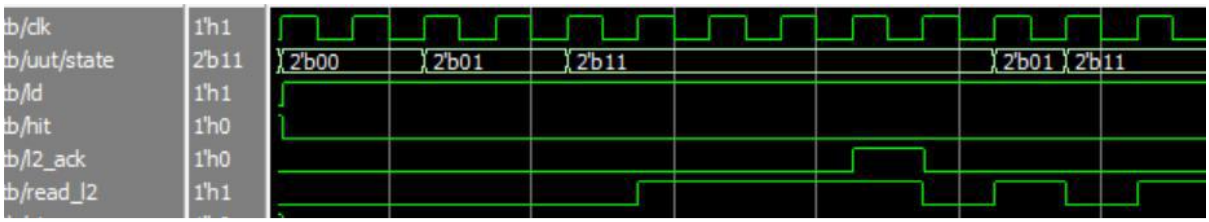
And output load_ready =1 to the CPU
Because the read signal continues, and hit, it goes from 00 → 01...
Until ld=0, st=1 becomes a write operation

- write hit



Prerequisite: write operation st=1, and hit=1 The state change is the same as the above read hit
Output signal write_l1=1

- read miss



- Prerequisite: Read missing ld=1, and hit=0, non-dirty block. L2_ack=0 from L2
  Output signal: read_l2, as long as it is in the Allocate state, it is equal to 1, and the number is taken from L2
- In the Allocate state for 8 cycles, the fetch is completed l2_ack=1
- Since the hit is determined by the tag and valid of the actual data, the Verilog I wrote does not involve the actual data

According to this test, the valid bit is always set to 0, so hit is always 0 and state is always in CompareTag Switch between and allocate without returning to Idle

- write miss

- Prerequisite: writing is missing st=1, and hit=0, it doesn't matter whether it is dirty or not, all write buffer. L2_ack=0 from L2
  Output signal: write_l2 write buffer
- State Idle → CompareTag (after discovering miss) → WriteBuffer → Idle

Conclusion

A cache controller was designed using the bottom-up approach and its simulation verification was carried out for 4 distinct situations.

References

1. E Aghdasi. Asynchmua aptc machim syntbrais using dam driven clocka. In EURODAC-I 992. V.
2. Akella md G. Gopahkrid" SH[LPA: a high-level rynthsis ayatem fa self-tin4 circuits. In ICCAD-1992.
3. E Brunvmd. Thc NSR w. In HICSS-26 (1993).
4. https://www.instructables.com/Design-of-a-Simple-Cache-Controller-in-VHDL/
5. https://www.wikipedia.org/