ECE401 PROJECT 2 REPORT
Submitted by – CHINMAI(29478051)

I. Introduction

Project required us to implement Instruction cache and Data cache in the MIPS pipelined processor. IF stage of the MIPS pipeline fetches instructions from Instruction cache and MEM stage of the pipeline fetchs data from Data cache. Design requirements and implementation of Instruction cache and Data cache has been discussed in this report.

II. Design Requirements

Cache size for Instruction cache and Data cache are as follows:

Specifications	Instruction Cache	Data Cache
Size	32KB	32KB
Associativity	Direct-Mapped	2-way
Block Size	32B	32B
Access Latency	1 cycle	1 cycle
Miss Penalty	10 cycles	10 cycles
Write Policies	N/A	Write-back. Write-allocated

Instruction Cache -

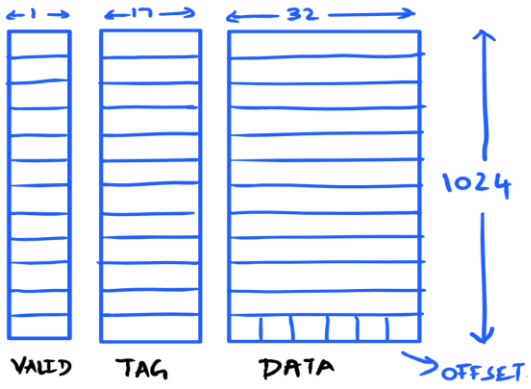
Direct Mapped associativity was used to implement Instruction cache. Direct Mapped cache consists of Tag, Valid and Data array as shown below. MEM stage sends address of the data it wants to read. Instruction cache uses index bits from the address and checks if the valid bit of the block is 1 and if the Tag of the block is same as the tag of the address. Offset bits from the address are used to access the specific byte from the block.

ADDRESS:

Tag[31:16]	Index[15:6]	Offset[5:0]

Direct Mapped cache of size 32KB
- 32B/line (8 words)
- 1 lines/set = 34B/set
- 32KB/32B= 1024sets

32-byte addresses:
- 5 bits: byte within line
- 17 bits: tag
- 10 bits: index



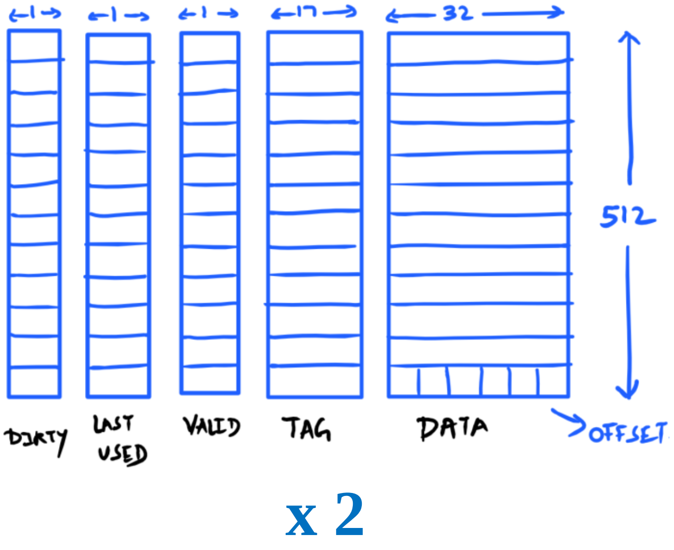
Data Cache –

2-way set associative was used to implement Data cache. 2-way set associative consists of Tag, 2 valid arrays, 2 Dirty arrays bits, Last used and 2 Data arrays as shown below. MEM stage sends address of the data it wants to read. Data cache uses index bits from the address and checks if the valid bit of the block is 1 before reading or writing data to the cache and if the Tag of the block is same as the tag of the address. Offset bits from the address are used to access the specific byte from the block.

ADDRESS:

Tag[31:15]	Index[14:6]	Offset[5:0]

2-way set associative cache of size 32KB
32B/line (8 words)
2lines/set = 64B/set
32KB/64B= 512sets
2lines/set*512sets=1024lines
32-byte addresses:
5 bits: byte within line
18 bits: tag
9 bits: index



III. Implementation

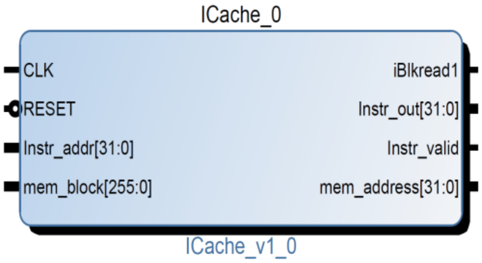
Instruction Cache –

Pin diagram given below represents the Instruction cache.

Cache HIT – Instr_addr is used to get the current index, tag and offset to access data from Instruction Cache. Instr_Valid gives if the instruction was valid or not. If it was a successful hit to the Cache Instr_out produces the correct instruction and Instr_Valid is 0.

Cache MISS – Instr_addr is used to get the current index, tag and offset to access data from Instruction Cache. If it was a cache Miss Cache sends iBlkread1 as 1 indicating the memory that it wants to read data from memory and also sends the missed block address via mem_address.

Memory sends the data via mem_block. Since it was a cache miss Instr_Valid is 1. Indicating the that this miss will be serviced after 10 cycles and hence all stages of the pipeline IF ID EXE MEM will be stalled. When the Cache miss gets serviced valid bit of the block will be set to 1.

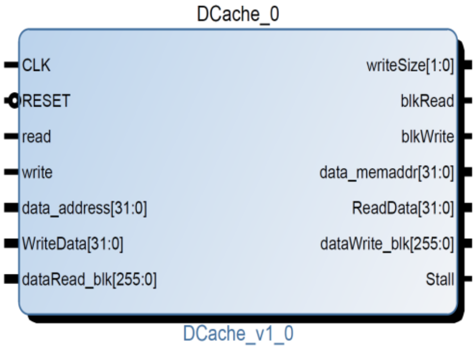


Data Cache-

Pin Diagram given below represents the Data cache used.

Cache HIT – data_address is used to get the current index, tag and offset to access data from data Cache. Stall gives if the data was valid or not. If it was a successful hit to the Cache ReadData produces the correct data on read and Stall is 0. For a write operation cache performs the write to location accessed using data_address and sets the dirty bit of this block as 1.

Cache MISS – data_address is used to get the current index, tag and offset to access data from Data Cache. If it was a cache Miss Cache sends iblkRead as 1 indicating the memory that it wants to read data from memory and also sends the missed block address via data_memaddr. Memory sends the data via data_Readblock. Since it was a cache Miss Stall is 1. Indicating that this miss will be serviced after 10 cycles and hence all stages of the pipeline IF ID EXE MEM will be stalled. Before performing the replacement of a block cache check is made if this block is dirty and the block is written to the main memory if the block was found to be dirty. blkWrite is set to 1 and block is sent via dataWrite_blk to the main memory. Once the blk is written to the main memory, cache miss is then serviced. The missed block is read from the memory.



Replacement Policy – To perform replacement of the block we maintain a Last_used array.
Before replacing the block a check is made to see which block was used last and replace the block that was not used recently

IV. Experimental Results

Performed Tests present in tests folder. The processor was successful for pipetest,airth, branchtest. Hence tests were followed to perform noio tests. However processor stopped functioning after 2470 cycles. Write back operations were not getting performed in the correct cycle. However few of the application do not pass the tests since Syscall hasn't been implemented.

Application	Instruction Count
noio	2081
file	96232
hello	
class	
sort	
fact12	
matrix	SYSCALL 4048
hanoi	
ical	
fib18	