Processor Architecture
Pipelined Processor

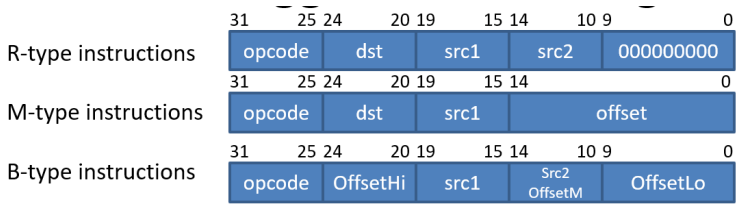
Final Report

Marc Igarza Mateo
Joel Sanchez Moreno

January 13, 2020

1 Processor Description

In this section, we detail the main characteristics of the implemented processor. The implemented Instruction Set Architecture (ISA) is based on MIPS as described on the project guidelines. There are 3 types of instructions and we can see the encoding in the following figure:



The R-type supported instructions are ADD, SUB, MUL, ADDI, SLL and SRL. The M-type instructions are memory access instructions on what we support LDB, LDW, STB and STW, and finally the B-type instructions are the branches instructions for what we have BEQ, BLT, BGT, BLE, BGE, BNE and JUMP. In addition, we included 3 more instructions in order to manage the exceptions, which are MOV, TLBWRITE and IRET encoded as M-type, B-type and B-type respectively.

The register file has 32 registers of 32 bits each and 4 special registers rm0, rm1 and rm2 for exceptions, and rm4 for the current privilege state of the machine (0 for users and 1 for supervisor). The processors boots from address 0x1000 with Supervisor mode, which means that virtual memory is not enabled, and the exceptions always force the core to jump to address 0x2000.

In this project we have two levels of memory hierarchical: the top one is the main memory, and the bottom one consists on the instruction and data cache which have 4 cache lines of 128 bits per cache line with an LRU replacement policy and two-way associative. The data cache has a Store Buffer with 8 entries which provides the core a better performance.
The latency access from the instruction and data cache to main memory is of 10 cycles, 5 to go to memory and 5 to return the data.

This processor has 5 states in the pipeline: Fetch, Decode, ALU or MUL, Cache and WriteBack (WB). The MUL instructions take 5 cycles on the MUL stage to compute the value. In the case of a MUL, the minimum pipeline is F, D, M1, M2, M3, M4, M5 and WB.

There is a Reorder Buffer of 8 entries that ensures the core has a better performance. The ALU and MUL stages are connected to the Reorder Buffer such that we have the full set of bypasses needed to speed up the execution.

The core also supports virtual memory, in order to do that we have an instruction TLB (iTlb) on the fetch stage and a data TLB (dTlb) on the cache stage. The Virtual Address (VA) size is 32 bits and the physical one is 20 bits, with pages of 4KB. The way the Physical Address (PA) is computed depends on the exception handler routine, which in our tests perform the next operation: PA = VA + 0x8000. As has been noted above, when we boot we are by default in Supervisor mode, where the virtual memory is disabled, so we don't have to look into the TLB when register rm4 is 1. When a TLB miss occurs, we save the state in rm0, rm1 and rm2 and we jump to the exception handler, which takes care of resolving the issue. The exception handler makes use of TLBWRITE, MOV and IRET instructions to resolve that exception and return to the source code.

It is important to note that when the core performs an IRET instruction to jump to the instruction that triggered the exception, it returns on User mode, which means that virtual memory is now enabled.

2 Performance Tests

The results we obtained executing the performance tests are the following:

- The buffer sum is executed in 1.975,5 ns
- Matrix Multiplication last 267.561.575 ns

3 Schematics

In the next figures we show a detailed schematic of each pipeline stage and then a global simplified schematic that show which part of the processor is used when we execute each type of instruction.

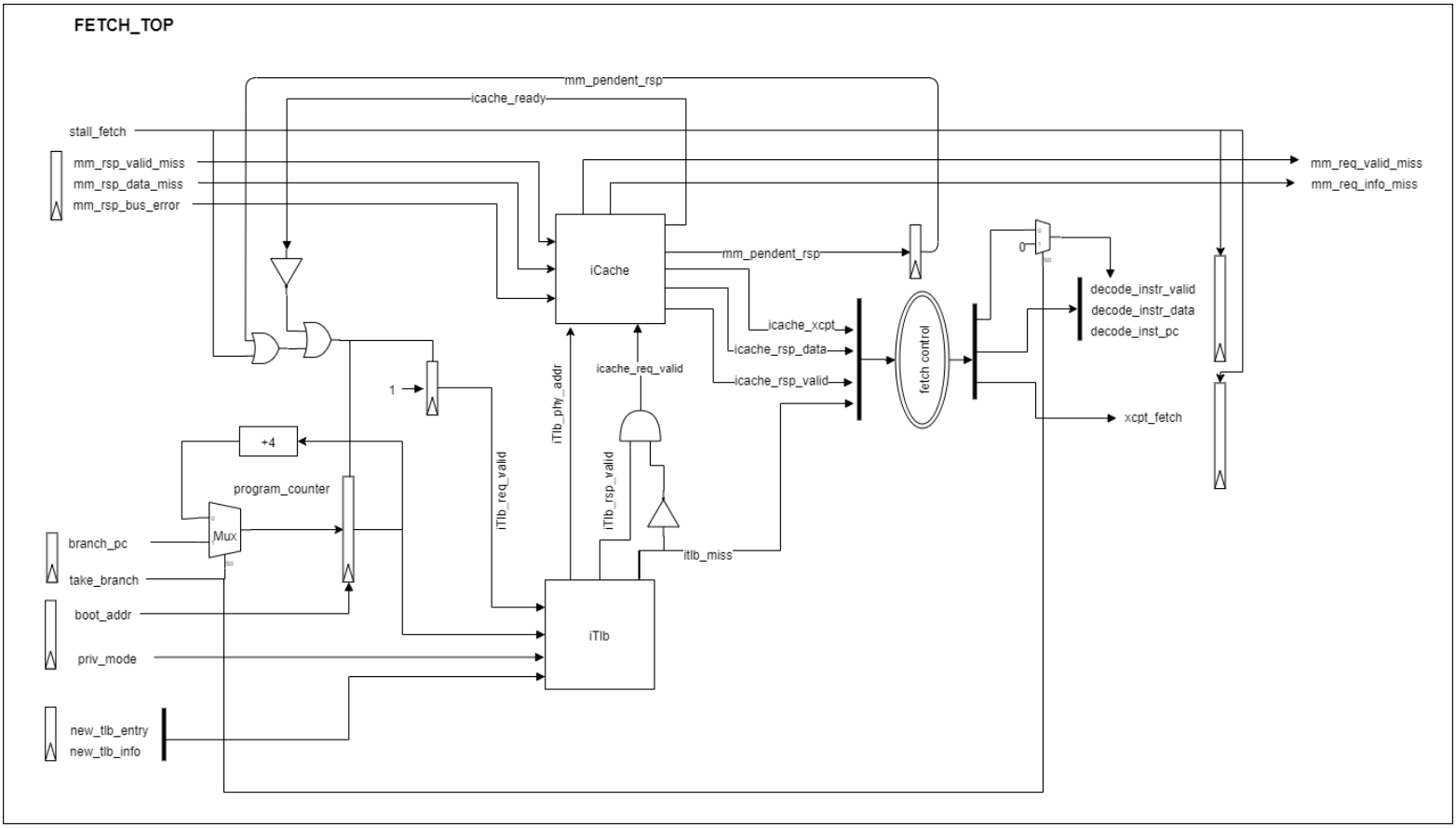


Figure 1: Fetch Stage

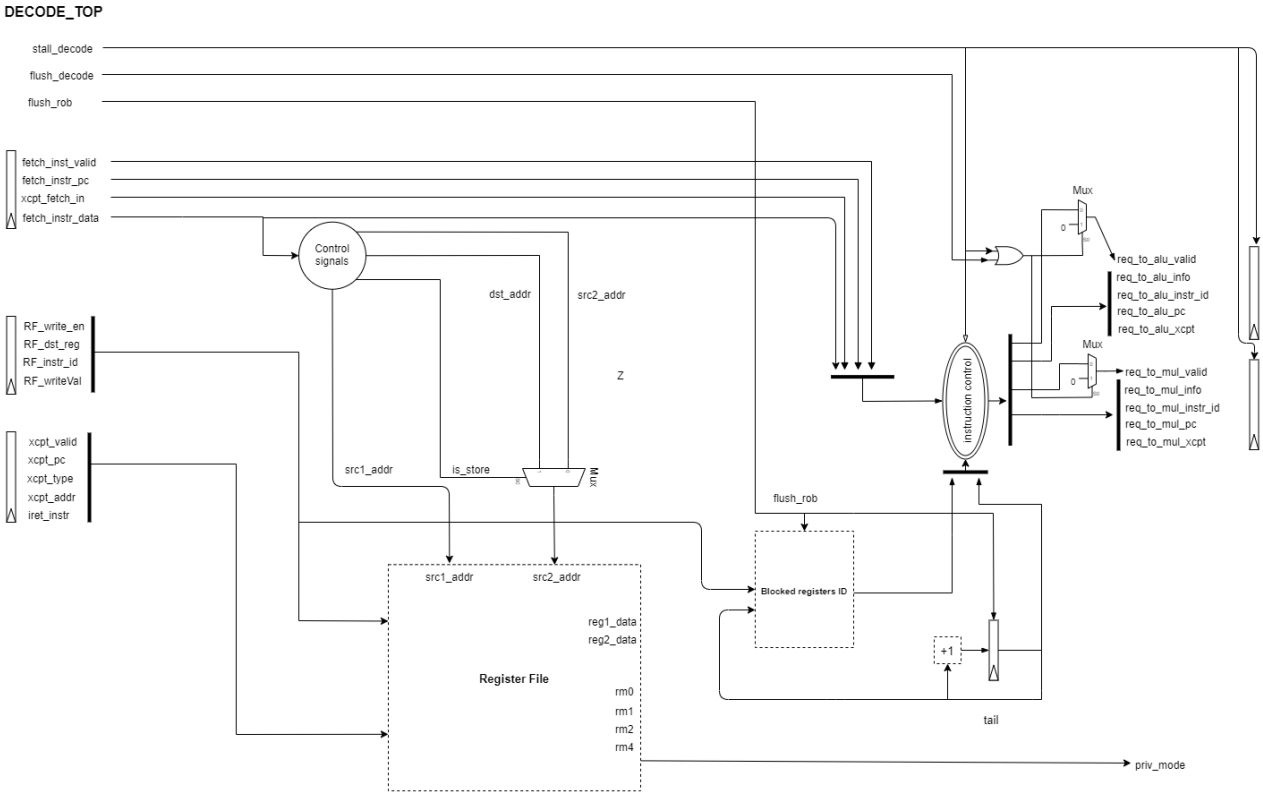


Figure 2: Decode Stage

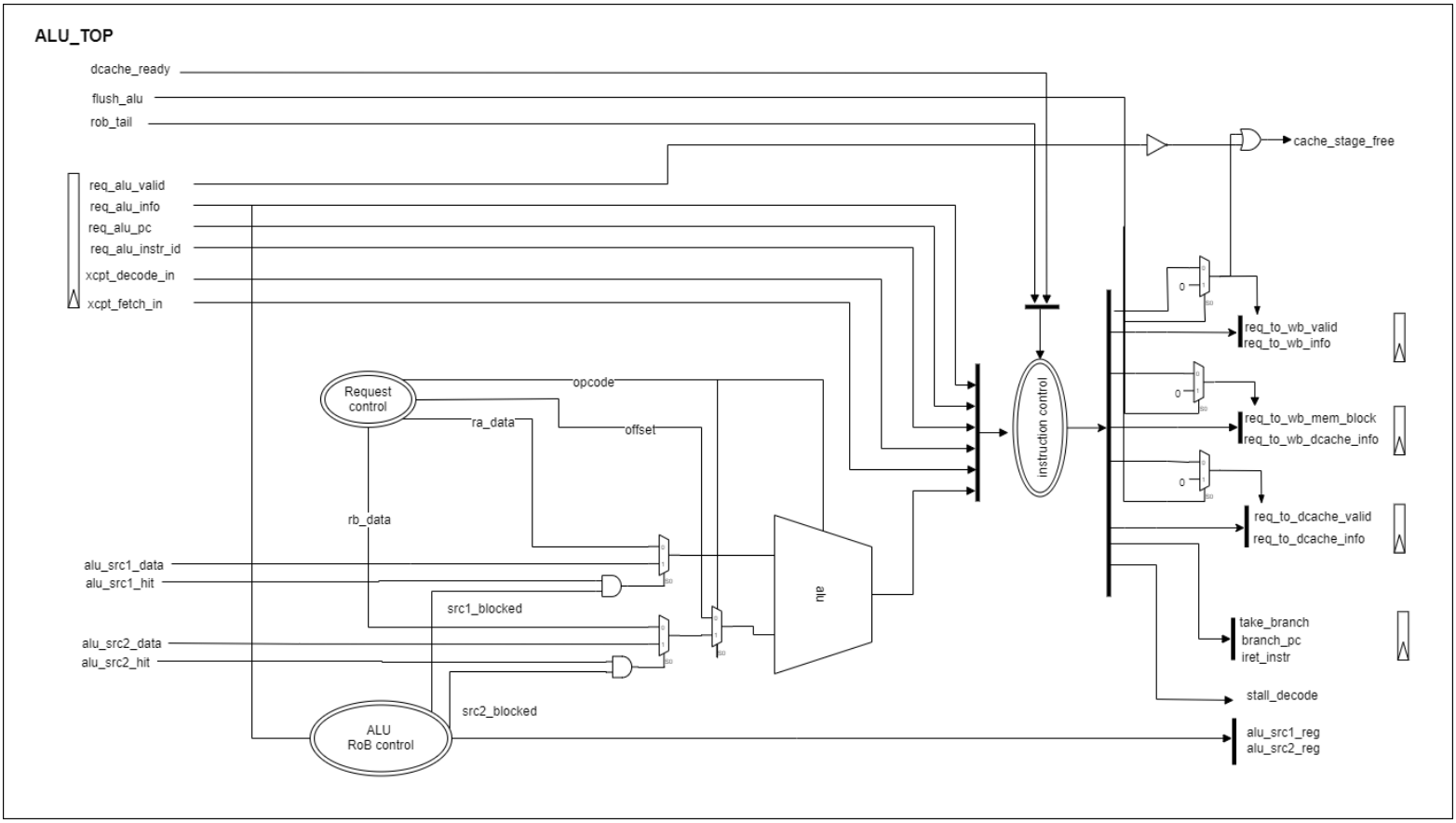


Figure 3: ALU Stage

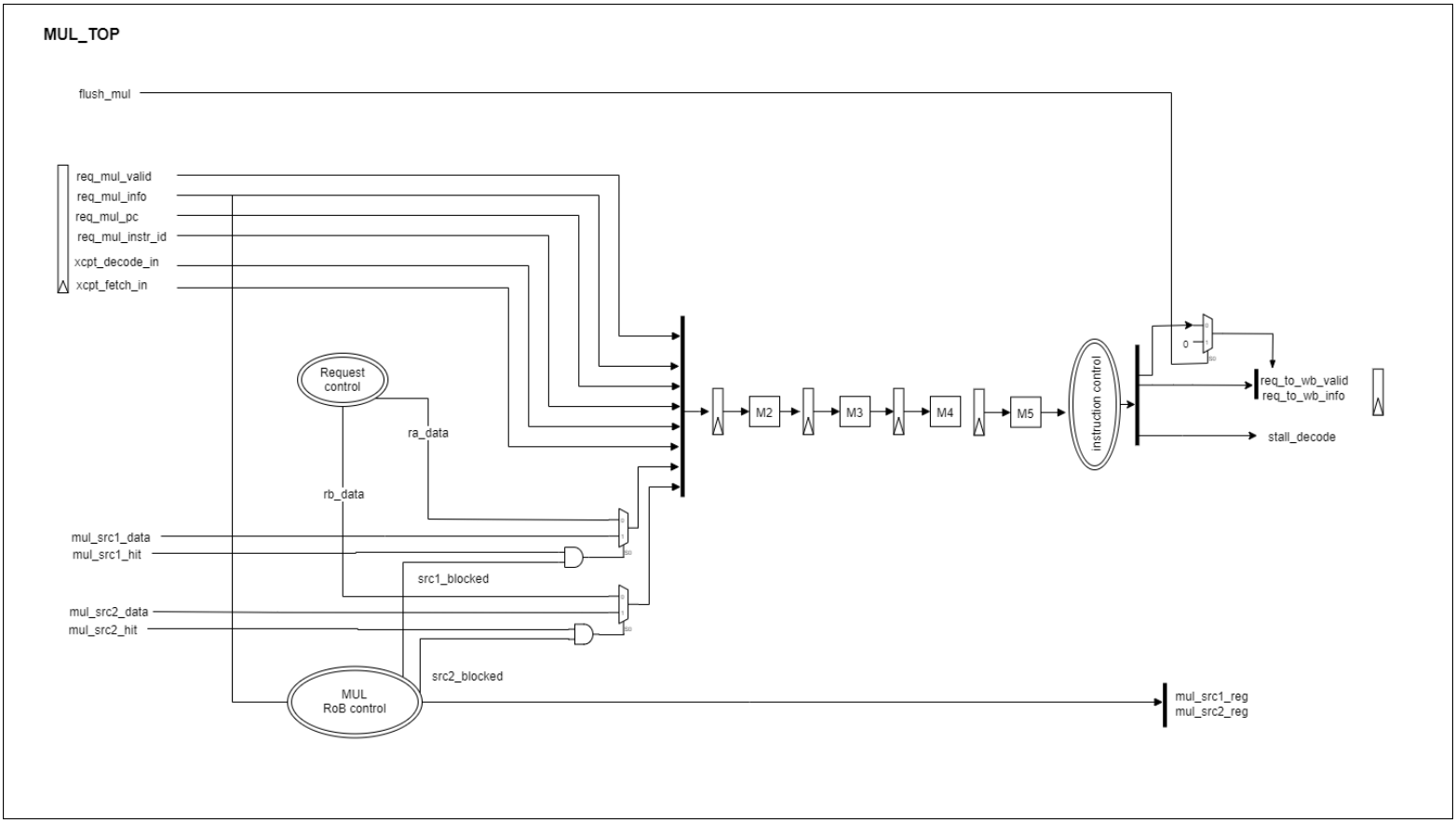


Figure 4: MUL Stage

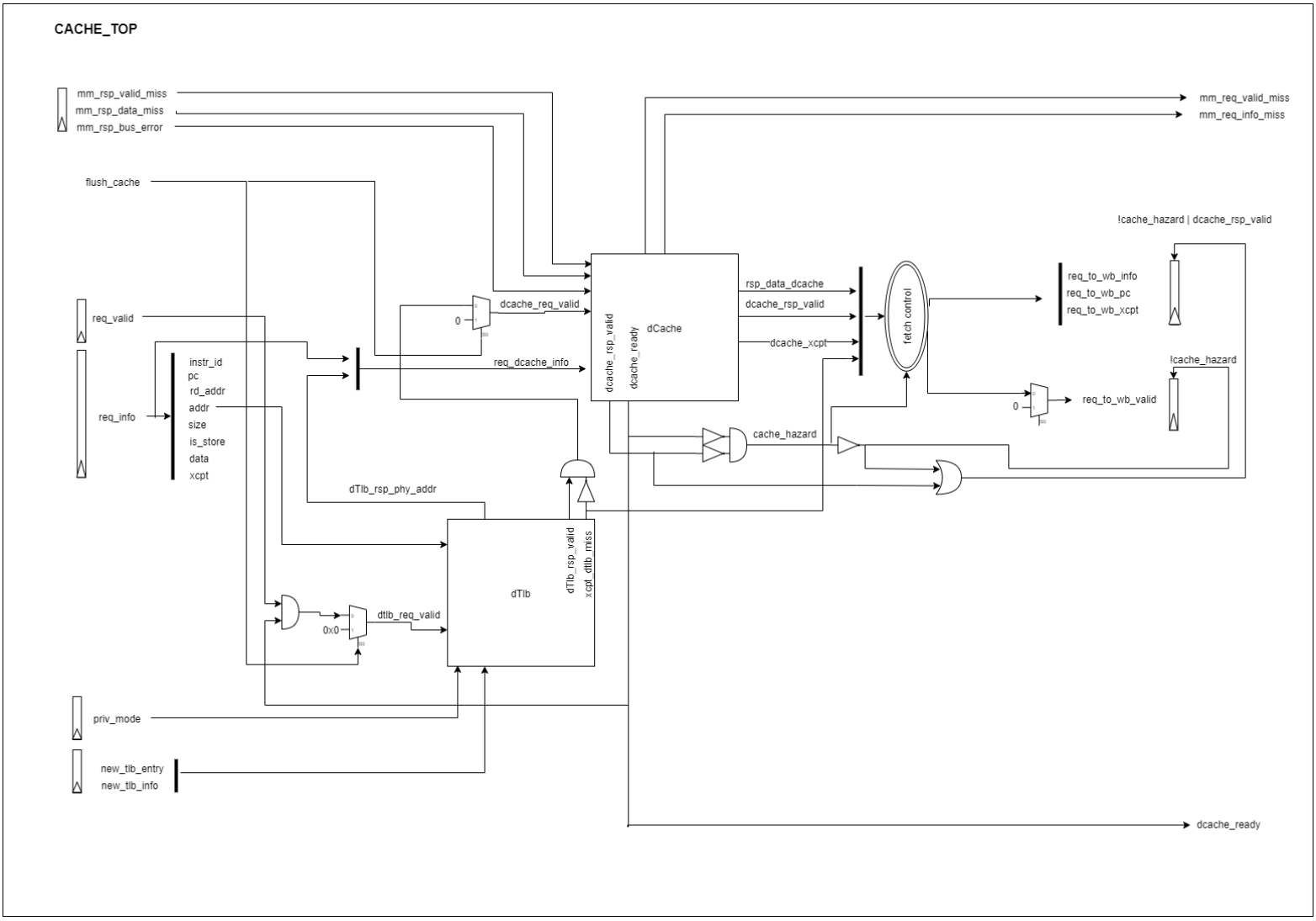


Figure 5: Cache Stage

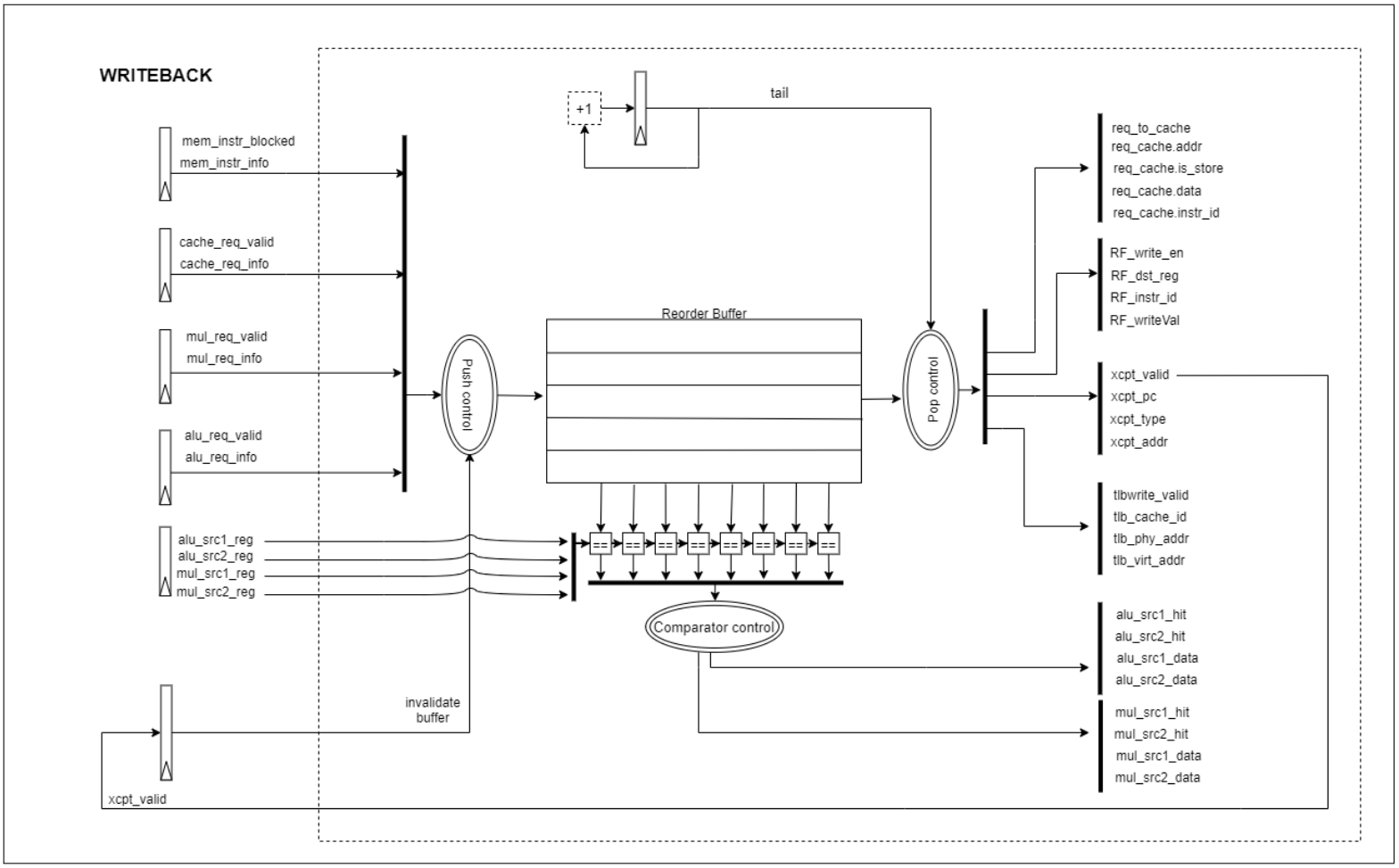


Figure 6: WriteBack Stage

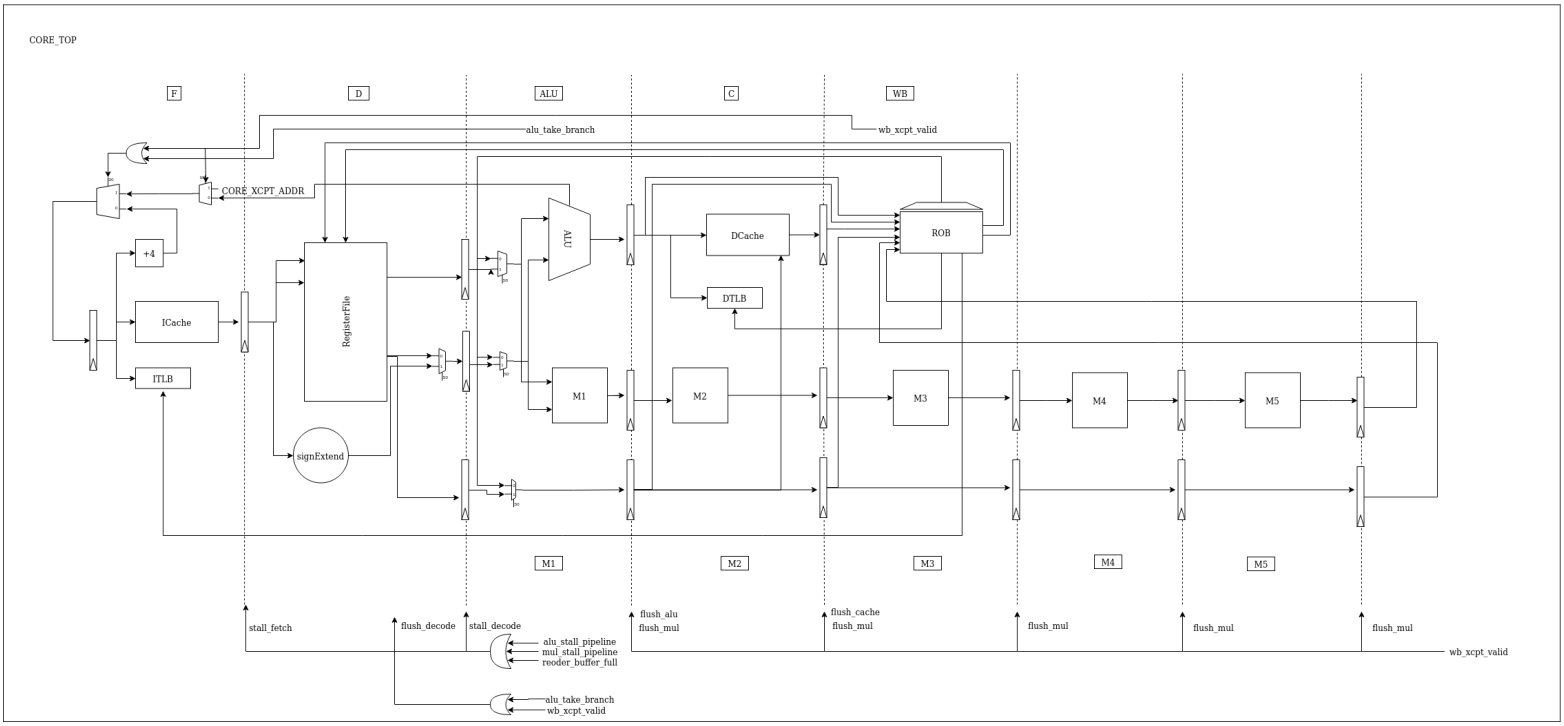


Figure 7: Full Pipeline

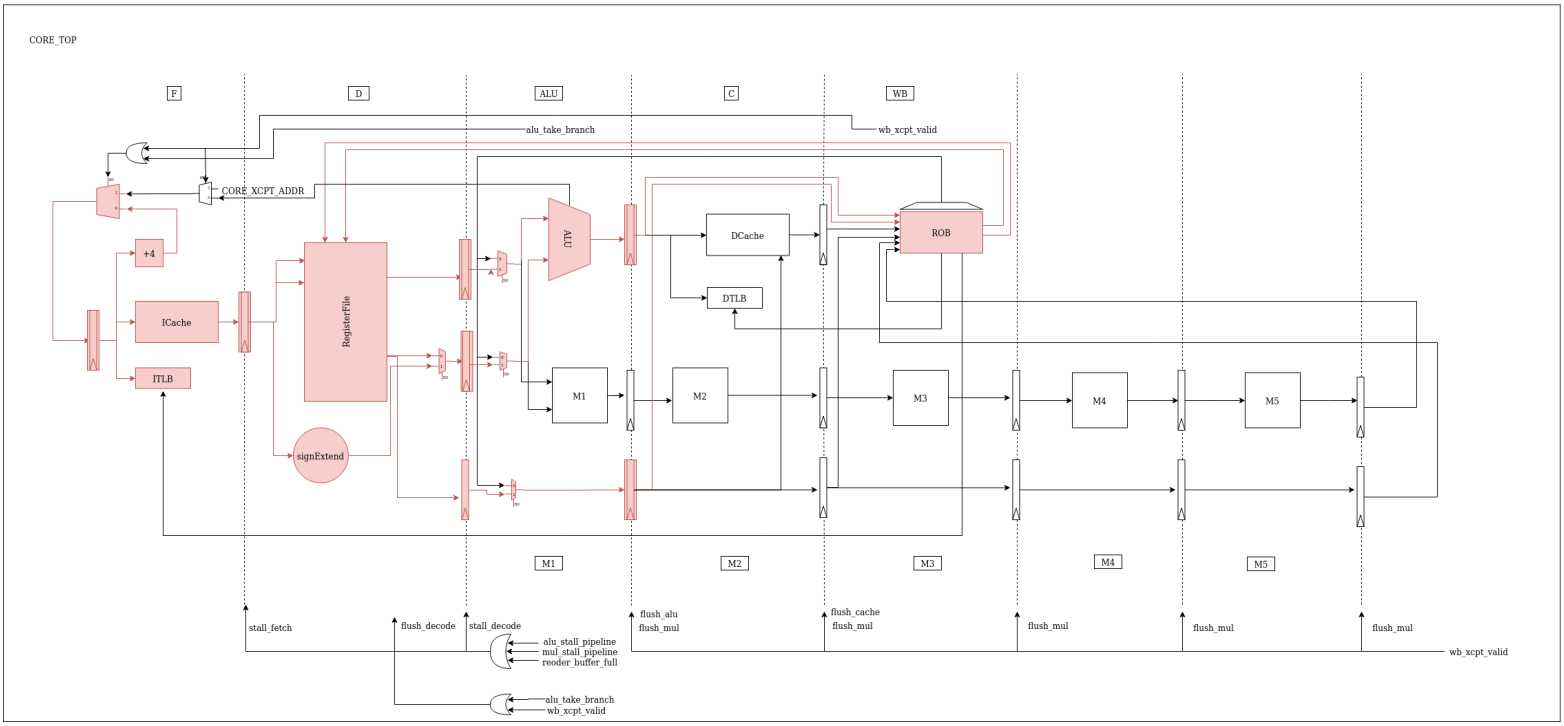


Figure 8: Full Pipeline - R-type Instructions

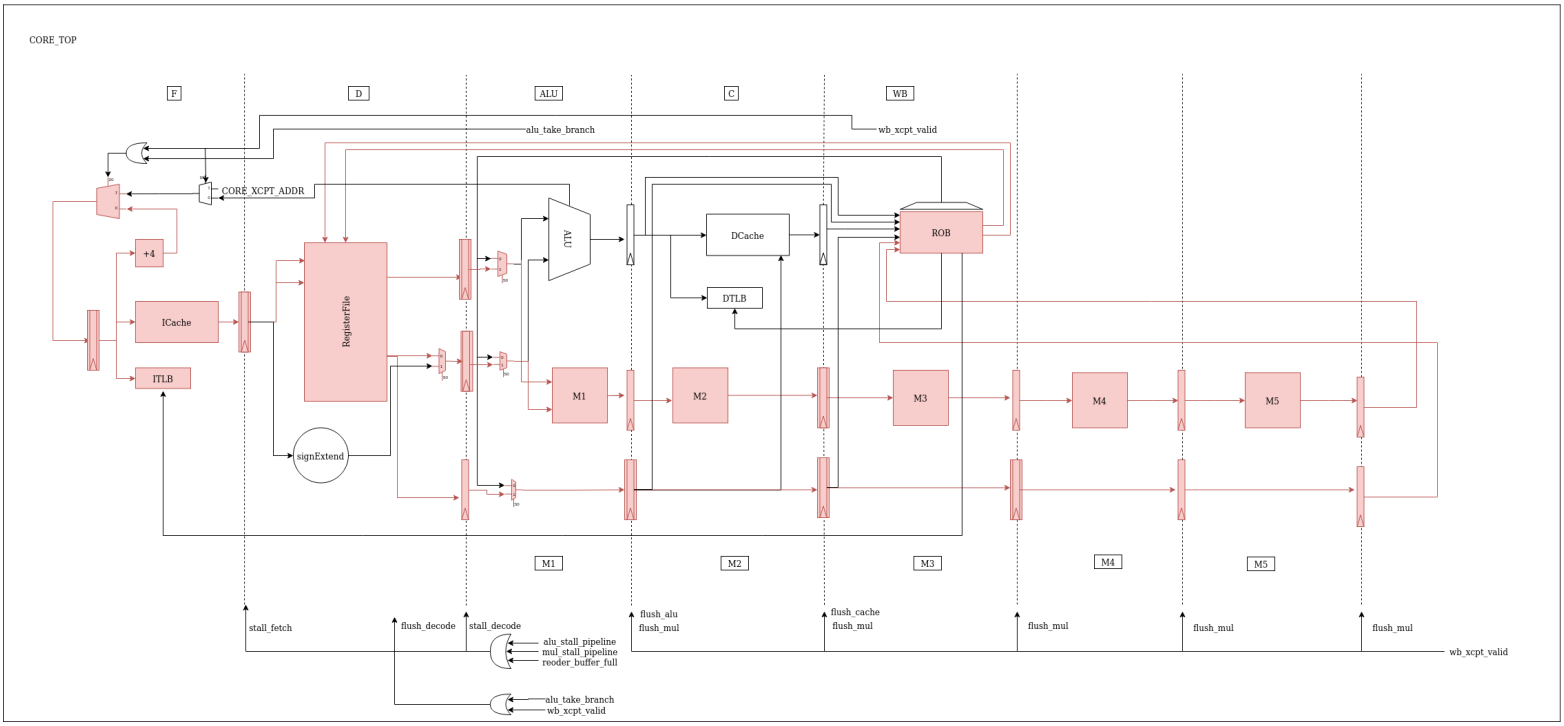


Figure 9: Full Pipeline - MUL Instruction

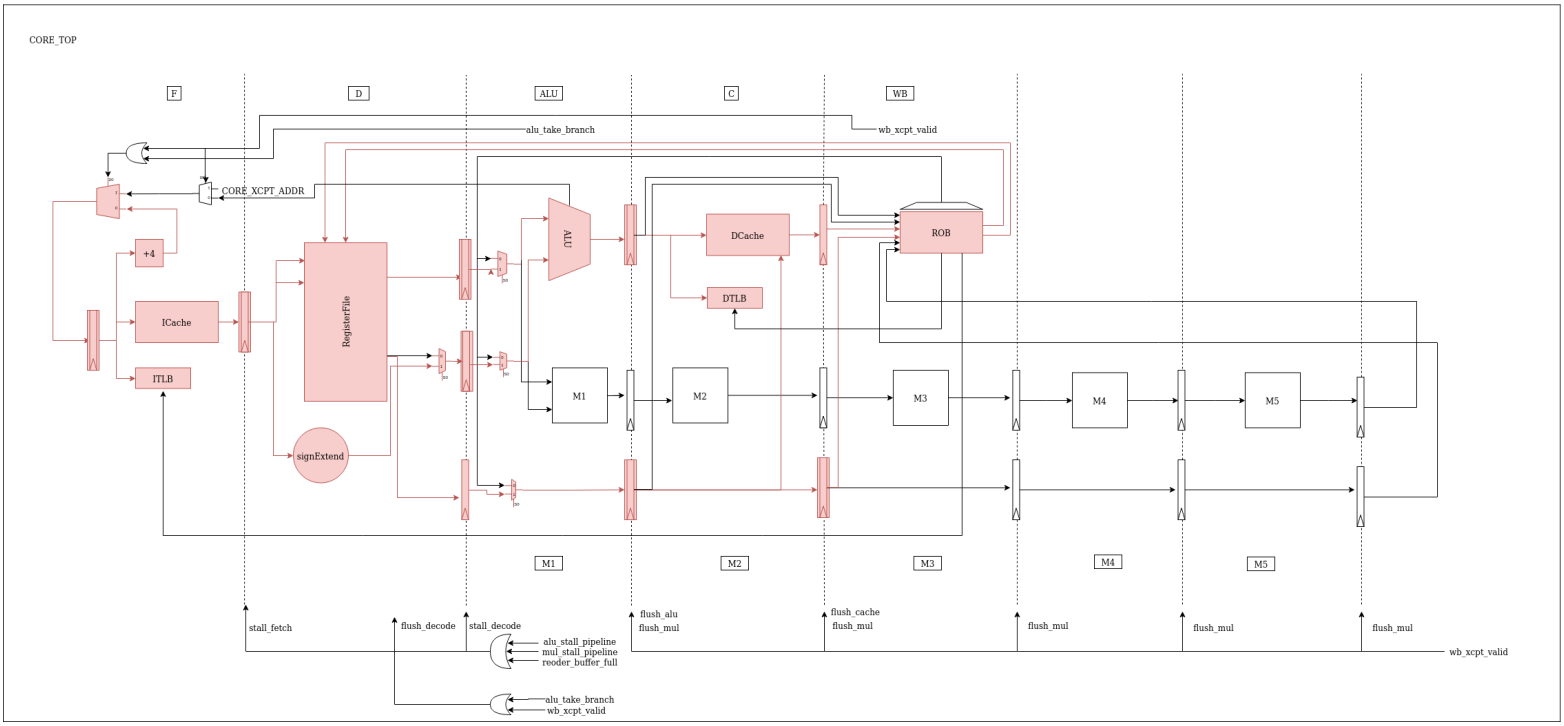


Figure 10: Full Pipeline - M-type Instructions

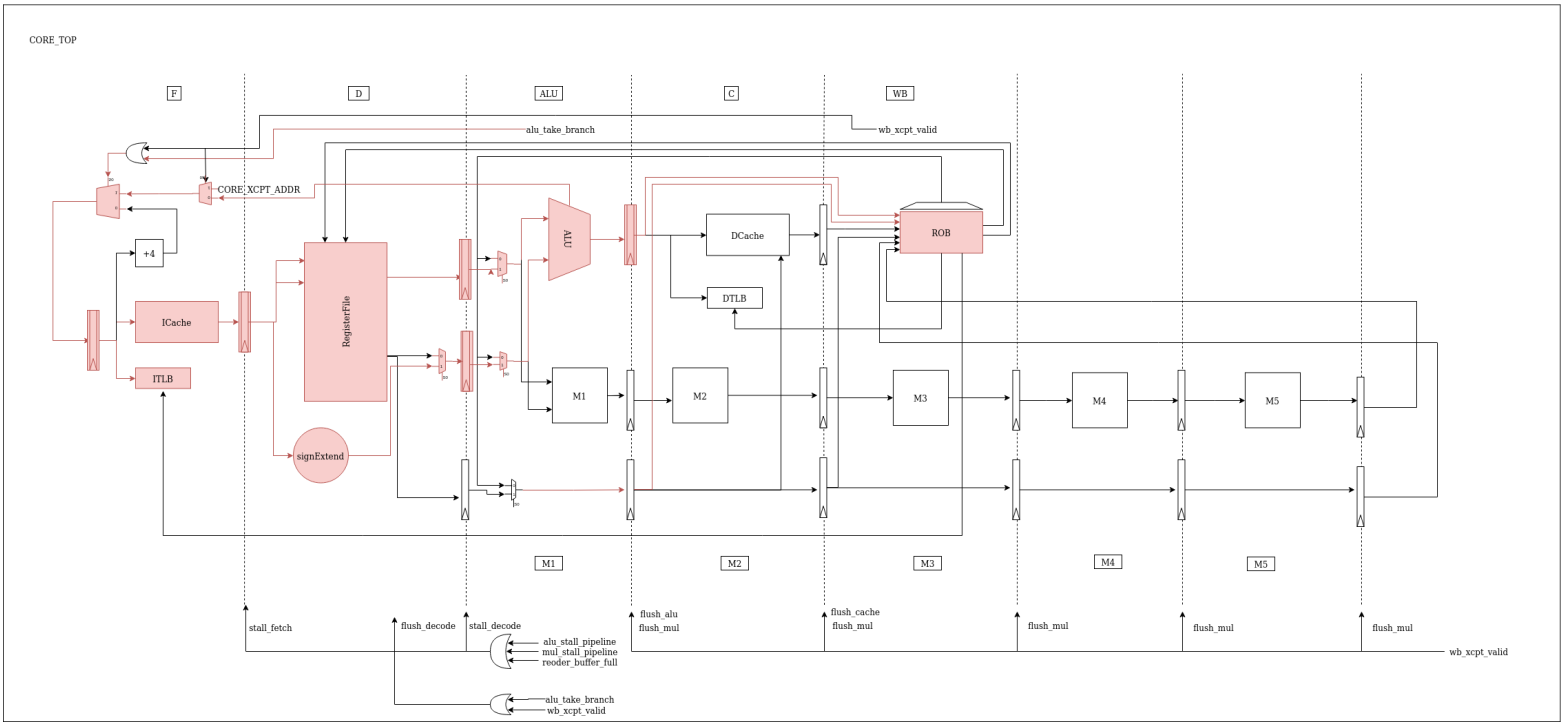


Figure 11: Full Pipeline - B-type Instructions