A diagram of a computer

AI-generated content may be incorrect.Micro Report

Hierarchical Datapath Diagram

Our CPU has 5 stages: Instruction Fetch (IF) (blue rectangle), Decode (ID)(green donut making sure to exclude FSM), Execute (EX)(Orange rectangle), Memory Access (MEM)(red rectangle), and Write Back (WB)(purple rectangle. At each boundary between stages, there are pipeline registers to hold relevant values and maintain instruction state across clock cycles. These are IF\_ID, ID\_EX, EX\_MEM, and MEM\_WB.

The instruction fetch stage uses a program counter (PC) module to feed addresses into the instruction memory. The instruction output is then passed through the IF\_ID register to separate it from the next cycle's fetch. In the decode stage, the decoder module interprets the instruction, extracting control signals and register specifiers. The register file reads the source operands, and the immediate generator handles sign-extension and immediate field extraction as needed.

Values from decode pass through ID\_EX, which forwards everything needed for the next stage. In execute, the ALU performs arithmetic or logical operations depending on the instruction. Immediate values have also been routed here directly from ID\_EX, making sure both register and immediate-type instructions are supported.

After execution, results and relevant control signals enter the EX\_MEM register before hitting the data memory module in the memory stage. This module handles loads and stores. The final pipeline register, MEM\_WB, captures the memory output or ALU result and feeds it back into the register file if required.

FSM

A diagram of a diagram

AI-generated content may be incorrect.The finite state machine (FSM) in this design controls the high-level operation of a simple CPU by sequencing through five key stages: Fetch, Decode, Execute, Memory, and Writeback. Each state in the FSM corresponds to one stage in the instruction cycle, and transitions follow a fixed order, looping continuously as instructions are processed.

On reset, the system initializes and moves directly into the Fetch state. In this stage, the CPU retrieves the instruction from memory using the current program counter (PC). Once the instruction is fetched, the FSM transitions to the Decode state.

In the Decode stage, the instruction’s opcode is examined to determine what operation needs to be performed. The control unit sets up the necessary signals based on this opcode. Notably, the actual control signals are generated in the decoder module independently of the FSM state — meaning the FSM is responsible only for sequencing, not for generating control signal values.

After decoding, the FSM proceeds to the Execute state. Here, the ALU performs the required arithmetic or logic operation. Depending on the instruction, it might use an immediate value or register operands, and the result may be used directly or passed to the next stage.

From Execute, the FSM always moves to the Memory stage. This is where data memory operations occur, if needed. For example, if the instruction is a load or store, this is when the data is read from or written to memory. If no memory operation is required (like in an ADD), the stage effectively does nothing for that cycle.

The final stage is Writeback, where results are written back into the register file, assuming the instruction requires it. Once complete, the FSM loops back to Fetch and begins the next instruction cycle. Overall, this FSM serves as the backbone for coordinating how each instruction flows through the processor.