PIPELINING

Basic concepts

- Pipelining is an implementation technique whereby multiple instructions are overlapped in execution
- Takes advantage of parallelism
- Key implementation technique used to make fast CPUs.
- A pipeline is like an assembly line.

In a computer pipeline, each step in the pipeline completes a part of an instruction. Like the assembly line, different steps are completing different parts of different instructions in parallel. Each of these steps is called a pipe stage or a pipe segment. The stages are connected one to the next to form a pipe—instructions enter at one end, progress through the stages, and exit at the other end.

- The throughput of an instruction pipeline is determined by how often an instruction exits the pipeline.

- The time required between moving an instruction one step down the pipeline is a processor cycle..

- If the starting point is a processor that takes 1 (long) clock cycle per instruction, then pipelining decreases the clock cycle time.

- Pipeline for an integer subset of a RISC architecture that consists of load-store word, branch, and integer ALU operations.

Every instruction in this RISC subset can be implemented in at most 5 clock cycles. The 5 clock cycles are as follows.

1. **Instruction fetch cycle (IF):**

   Send the program counter (PC) to memory and fetch the current instruction from memory. PC=PC+4

2. **Instruction decode/register fetch cycle (ID):**

- Decode the instruction and read the registers.
- Do the equality test on the registers as they are read, for a possible branch.
- Compute the possible branch target address by adding the sign-extended offset to the incremented PC.

- Decoding is done in parallel with reading registers, which is possible because the register specifiers are at a fixed location in a RISC architecture, known as fixed-field decoding

3. Execution/effective address cycle (EX):

The ALU operates on the operands prepared in the prior cycle, performing one of three functions depending on the instruction type.

- Memory reference: The ALU adds the base register and the offset to form the effective address.

- Register-Register ALU instruction: The ALU performs the operation specified by the ALU opcode on the values read from the register file

- Register-Immediate ALU instruction: The ALU performs the operation specified by the ALU opcode on the first value read from the register file and the sign-extended immediate.

4. Memory access(MEM):

- If the instruction is a load, memory does a read using the effective address computed in the previous cycle.

- If it is a store, then the memory writes the data from the second register read from the register file using the effective address.

5. Write-back cycle (WB):

Register-Register ALU instruction or Load instruction: Write the result into the register file, whether it comes from the memory system (for a load) or from the ALU (for an ALU instruction).

PIPELINED DATA PATH AND CONTROL

The Classic Five-Stage Pipeline for a RISC Processor

Each of the clock cycles from the previous section becomes a pipe stage—a cycle in the pipeline. Each instruction takes 5 clock cycles to complete, during each clock cycle the hardware will initiate a new instruction and will be executing some part of the five different instructions.

3 observations:

1. Use separate instruction and data memories, which implement with separate instruction and data caches.

2. The register file is used in the two stages: one for reading in ID and one for writing in WB, need to perform 2 reads and one write every clock cycle.

3. Does not deal with PC, To start a new instruction every clock, we must increment and store the PC every clock, and this must be done during the IF stage in preparation for the next instruction.

To ensure that instructions in different stages of the pipeline do not interfere with one another. This separation is done by introducing pipeline registers between successive stages of the pipeline, so that at the end of a clock cycle all the results from a given stage are stored into a register that is used as the input to the next stage on the next clock cycle.