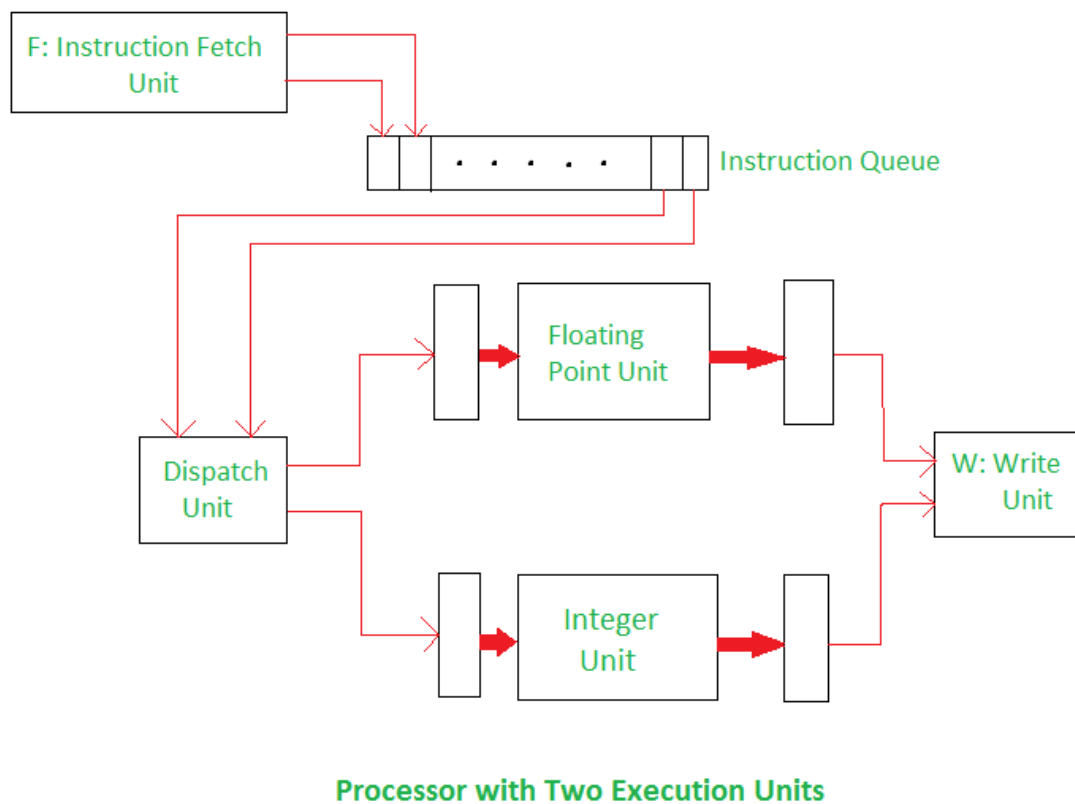


Superscalar Architecture

A more aggressive approach is to equip the processor with multiple processing units to handle several instructions in parallel in each processing stage. With this arrangement, several instructions start execution in the same clock cycle and the process is said to use multiple issue. Such processors are capable of achieving an instruction execution throughput of more than one instruction per cycle. They are known as 'Superscalar Processors'.



In the above diagram, there is a processor with two execution units; one for integer and one for floating point operations. The instruction fetch unit is capable of reading the instructions at a time and storing them in the instruction queue. In each cycle, the dispatch unit retrieves and decodes up to two instructions from the front of the queue. If there is one integer, one floating point instruction and no hazards,

both the instructions are dispatched in the same clock cycle.

Advantages of Superscalar Architecture :

- The compiler can avoid many hazards through judicious selection and ordering of instructions.
- The compiler should strive to interleave floating point and integer instructions. This would enable the dispatch unit to keep both the integer and floating point units busy most of the time.
- In general, high performance is achieved if the compiler is able to arrange program instructions to take maximum advantage of the available hardware units.

Disadvantages of Superscalar Architecture :

- In a Superscalar Processor, the detrimental effect on performance of various hazards becomes even more pronounced.
- Due to this type of architecture, problem in scheduling can occur.

Very Long Instruction Word (VLIW) Architecture

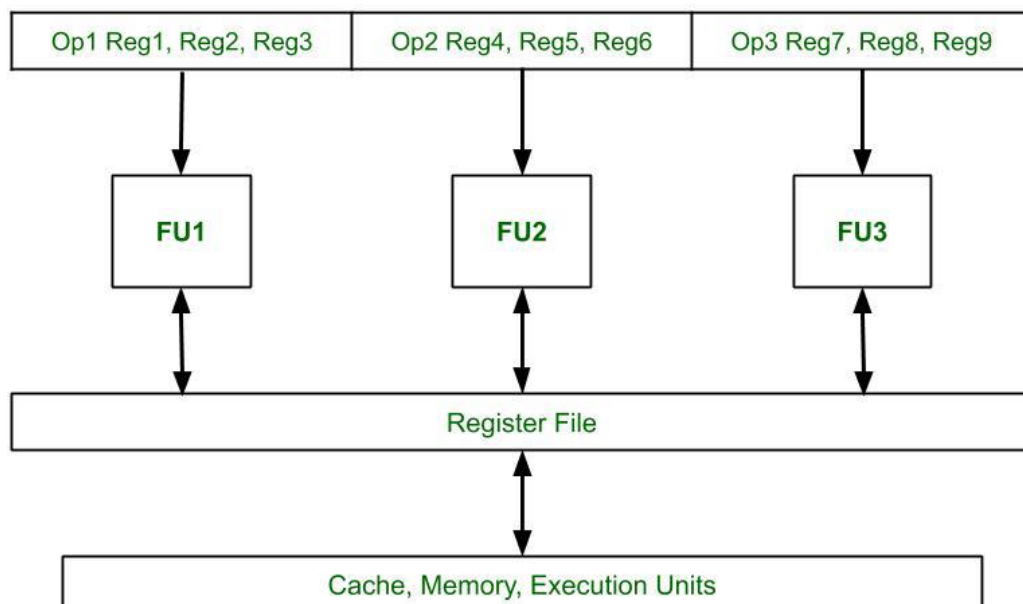
The limitations of the Superscalar processor are prominent as the difficulty of scheduling instruction becomes complex. The intrinsic parallelism in the instruction stream, complexity, cost, and the branch instruction issue get resolved by a higher instruction set architecture called the **Very Long Instruction Word (VLIW)** or **VLIW Machines**. VLIW uses Instruction Level Parallelism, i.e. it has programs to control the parallel execution of the instructions.

In other architectures, the performance of the processor is improved by using either of the following methods: pipelining (break the instruction into subparts), superscalar processor (independently execute the instructions in different parts of the processor), out-of-order-execution (execute orders differently to the program) but each of these methods add to the complexity of the hardware very much. VLIW

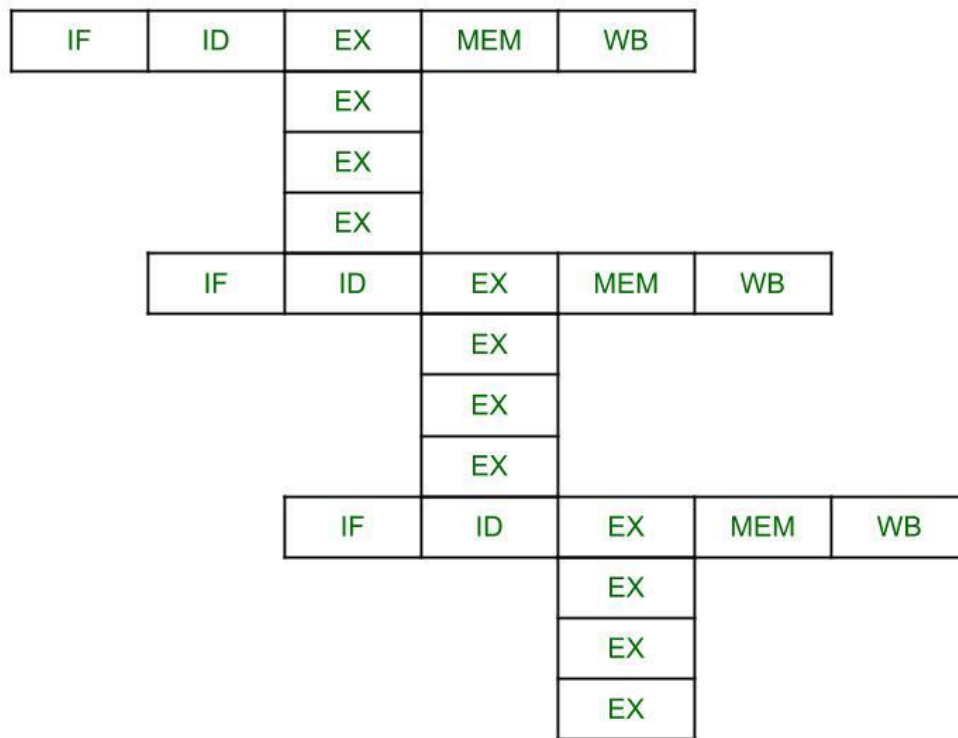
Architecture deals with it by depending on the compiler. The programs decide the parallel flow of the instructions and to resolve conflicts. This increases compiler complexity but decreases hardware complexity by a lot.

Features :

- The processors in this architecture have multiple functional units, fetch from the Instruction cache that have the Very Long Instruction Word.
- Multiple independent operations are grouped together in a single VLIW Instruction. They are initialized in the same clock cycle.
- Each operation is assigned an independent functional unit.
- All the functional units share a common register file.
- Instruction words are typically of the length 64-1024 bits depending on the number of execution unit and the code length required to control each unit.
- Instruction scheduling and parallel dispatch of the word is done statically by the compiler.
- The compiler checks for dependencies before scheduling parallel execution of the instructions.



BlockDiagramofVLIWArchitecture



IF: Instruction Fetch ID: Instruction Decode EX: Execute MEM: Memory WB: Write Back

Time Space Diagram of VLIW Processor where 4 instructions are executed in parallel in a single instruction word

