Instruction level parallelism (ILP)

ILP is used to refer to the architecture in which multiple operations can be performed parallely in particular process, with its own set of resources- address space, registers, identifiers state and program counters.

What is Instruction level parallelism?

Instruction level parallelism can also appear explicitly in the instruction set. VLIW (very low long Instruction word) machine have instructions that can issue multiple operations in parallel. The Intel IA64 is a well known example of such an architecture. All high-performance, general purpose micro processors also include to generate code automatically for such machines from sequential programs. Such a compiler hides from the programmers the details of finding parallelism in a program, distributing the

the computation across the machine and minimizing synchronization and communication among the processors.

Example: Example of architecture that exploit ILP are VLIWs and superscalar Architecture. ILP processors have the same execution hardware as RISC processors. The machines without ILP have complex hardware which is hard to implement. A typical ILP allows multiple-cycle operations to be pipelined.

Instruction level parallelism (ILP) Architecture

ILP is achieved when multiple operations are performed in a single cycle, which is done by either executing them simultaneous or by utilizing gaps between two successive operations that are created due to the latencies. now the decision of when to execute an operation depends largely on the compiler rather than the hardware. However the extent of the compilers control depends on the type of ILP architecture where information regarding parallelism given by the compiler to hardware via the program varies.

Classification of ILP Architectures:

Sequential Architectures: Here the program is not expected to explicitly convey any information regarding parallelism to hardware like superscalar architecture.

Dependence Architecture: Here the program explicitly mentions information regarding dependencies between operation like data flow architecture.

Independence Architecture: Here programme gives information regarding which operations are independent of each other so that they can be executed instead of the inops.

Advantage of Instruction level parallelism:

* Improved performance

* Efficient Resource utilization
* Reduced Instruction dependency
* Increased throughput.

Disadvantages of Instruction - level parallelism:

* Increased complexity
* Instruction overhead
* Data dependency
* Reduced Energy Efficiency