1. **Superscalar (e.g., Intel Pentium, AMD Athlon)—multiple instructions in the same stage of execution in duplicate pipeline hardware**

   Alternatively, several instructions in the “execute” stage on different functional units.

   **Conceptual Depiction of Superscalar Processing**

   - **Instruction 1**
   - **Instruction 2**
   - **Instruction 3**
   - **Instruction 4**
   - **Instruction 5**
   - **Instruction 6**

---

**Pentium 4 Processor**

- Outer CISC shell (“old”/pre-RISC x86 architecture) with inner RISC core
- Inner RISC core pipeline at least 20 stages
- Some micro-ops require multiple execution stages

**Pentium 4 Operation**

- Fetch x86 (CISC) instructions from memory in order of static program
- Translate each x86 instruction into one or more fixed length RISC instructions, called micro-operations
- Execute RISC-like micro-ops on superscalar pipeline
  - micro-ops may be executed out of order causing new data dependences: WAR and WAW
- Commit results of micro-ops to register set in original x86 program flow order

---

**Lecture 29 - 1**

- Trace cache (L1 cache) stores recently executed micro-ops
- BTB uses dynamic branch prediction (a BHT) (4-bits used via Yeh’s algorithm)
- Static prediction used if not in BTB

---

**Lecture 29 - 2**

- b) Trace cache fetch
- a) Fetch 64 bytes of Pentium 4 (CISC) instructions from L2 cache and decode instruction boundaries and translates Pentium 4 (CISC) instructions into micro-op’s (RISC)

---

**Lecture 29 - 3**

- d) Drive delivers decoded instructions from the trace cache to the rename/allocate module
- c) Pulls micro-ops from cache (or ROM microprogrammed control unit for very complex instructions) in program sequence order

---

**Lecture 29 - 4**

- Pentium 4 Processor
  - Outer CISC shell (“old”/pre-RISC x86 architecture) with inner RISC core
  - Inner RISC core pipeline at least 20 stages
  - Some micro-ops require multiple execution stages
Out-of-Order Execution Logic:

- ROB entry contains: state, memory address of generating instruction, micro-op, renamed register.
- Two FIFO queues to hold micro-ops until there is room in the scheduler.
- One queue holds load or stores micro-ops.
- One queue holds the remaining non-memory micro-ops.
- Queues can operate at different speeds.
- Allocate - allocates resources needed for execution:
  - Stalls pipeline if a resource (e.g., register) is unavailable.
  - Uses a reorder buffer (ROB) to store information about a micro-op as it executes.
  - One of 128 integer or float registers for the result and/or one of 48 load buffers or one of 24 store buffers.
  - An entry in one of the two micro-op queues.

Computes flags - N, Z, C, V to use as input to the branches.

- If branch outcome does not match prediction, remove micro-ops from the pipeline.
- Provide proper branch destination to the BTB which restarts the whole pipeline from the correct target address.

Up to 6 micro-ops can be dispatched per cycle.

Scheduler retrieves micro-ops from queues for dispatching/issuing for execution if all operands and execution unit are available.