Computing at the Nanosacle
Lecture 01

(The state of state)

Impact of modern processing principles
(Lots of "state")

- *User*:
  - state used for application execution
- *Supervisor*:
  - state used to manage user state
- *Machine*:
  - state that configures the system
- *Transient*:
  - state used during instruction execution
- *Access-Enhancing*:
  - state used to simplify translation of other state names
- *Latency-Enhancing*:
  - state used to reduce latency to other state values

Impact of modern processing principles
(User State)

- *Purpose*:
  - State used for application execution
- *Example*:
  - Architectural Registers, PC, Flags
- *Size*:
  - Depends on ISA
- *Structure*:
  - Depends on ISA
- *Persistence*:
  - Long
- *Volatility*:
  - Changes on per instruction basis
- *Visibility*:
  - Completely

Impact of modern processing principles
(Supervisor State)

- *Purpose*:
  - State used to manage user state
- *Example*:
  - Page Maps, Task Controls
- *Size*:
  - Relatively small & independent of ISA
- *Structure*:
  - Primarily pointer registers
- *Persistence*:
  - Lifetime of a process
- *Volatility*:
  - Small
- *Visibility*:
  - Primarily OS

Impact of modern processing principles (Machine State)

- *Purpose:*
  - Configure System, Support Debug & Test
- *Example*:
  - Boot-time setup, Breakpoint set, Performance counters
- *Size*:
  - Relatively independent of ISA
- *Structure*:
  - Somewhat complex
- *Persistence*:
  - As long as powered up
- *Volatility*:
  - relatively low
- *Visibility*:
  - Primarily boot & OS

Impact of modern processing principles (Transient State)

- *Purpose:*
  - Improve performance
- *Example*:
  - Pipe latches, BHT, ROB, ...
- *Size*:
  - Dependent on pipe depth, issue width & order
- *Structure*:
  - Very complex
- *Persistence*:
  - Cycle to thread timeframe
- *Volatility*:
  - very high
- *Visibility*:
  - invisible to software

Impact of modern processing principles (Access-Enhancing State)

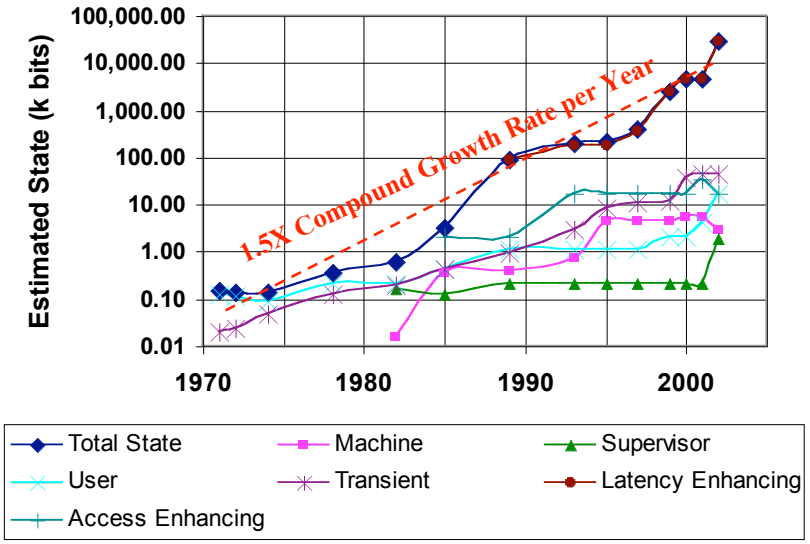
- *Purpose*:
  - State used to simplify translation of other state names
- *Example*:
  - Segment mapping regs, TLB
- *Size*:
  - Medium large
- *Structure*:
  - Fairly regular
- *Persistence*:
  - Cycle to process timeframe
- *Volatility*:
  - moderate
- *Visibility*:
  - Somewhat visible to OS

Impact of modern processing principles (Latency Enhancing State)

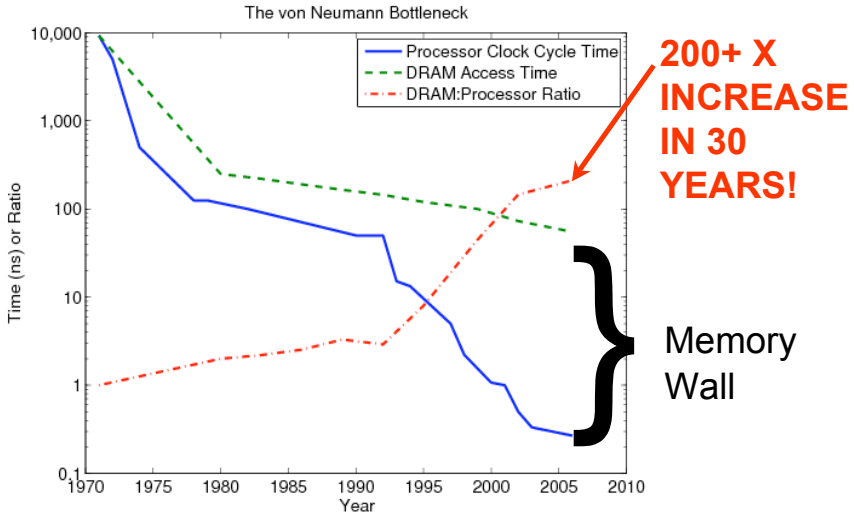
- *Purpose*:
  - State used to reduce latency to other state values
- *Example*:
  - Caches, Vector Registers
- *Size*:
  - Very large
- *Structure*:
  - Very regular
- *Persistence*:
  - process
- *Volatility*:
  - moderate
- *Visibility*:
  - through effects on performance

For dramatic effect, let's pause and think about how much state is associated with each kind of state…

Impact of modern processing principles
(Total State vs. Time)



Impact of modern processing principles
(Why so much latency enhancing state?)



Memory Hierarchies
(The principle of locality…)

- …says that most programs don't access all code or data uniformly
  - i.e. in a loop, small subset of instructions might be executed over and over again…
  - …& a block of memory addresses might be accessed sequentially…
- This has lead to "memory hierarchies"
- Some important things to note:
  - Fast memory is expensive
  - Levels of memory usually smaller/faster than previous
  - Levels of memory usually "subset" one another
    - All the stuff in a higher level is in some level below it

Memory Hierarchies
("always reuse a good idea")