

# MPSoC Clock and Power Challenges

Olivier Franza

Massachusetts Microprocessor Design Center

Digital Enterprise Group

Intel Corporation

July 14<sup>th</sup>, 2005



Copyright © 2005 Intel Corporation

Other names and brands may be claimed as the property of others



# Outline

- Current Challenges
- Industry Examples
- Future Directions
- Summary



# Current Challenges

## Definitions

- Power

- The rate at which work is done, expressed as the amount of work per unit of time in Watts
- In a microprocessor:

$$P = \alpha C V^2 F + V I_{\text{leak}}$$

$\alpha$  activity factor

C switching capacitance

V power supply voltage

F clock frequency

$I_{\text{leak}} \sim \exp(-qV_t/kT)$  leakage current

$V_t$  threshold voltage

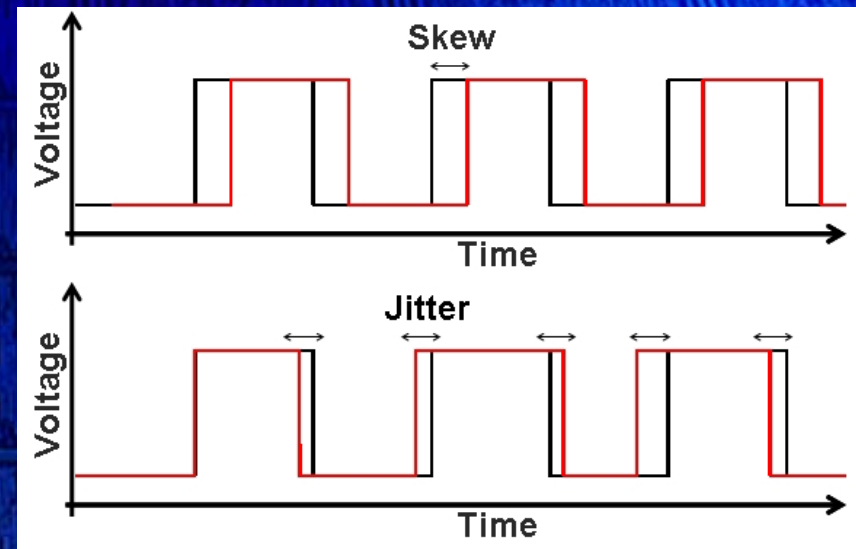
- The lower, the better...
- Clock is a key consumer of microprocessor power



# Current Challenges

## Definitions

- Clock
  - Source of regularly occurring pulses used to measure the passage of time
  - Heartbeat of synchronous digital systems: stability, regularity, accuracy, repeatability, and reliability all highly important
- Skew
  - Spatial inaccuracy of same clock edge arriving at various locations
- Jitter (cycle-to-cycle)
  - Temporal inaccuracy of successive clock edges arriving at the same location





# Current Challenges

## Objectives

- Ideal clock scenario
  - Zero skew, zero jitter
  - Perfect duty cycles
  - Short sharp rise and fall times, rail-to-rail signal
  - Low power consumption
  - Infinite frequency granularity, instantaneous dynamic frequency and voltage control
- In reality
  - Non-zero skew, non-zero jitter
  - Non-perfect variable duty cycle
  - Noticeable rise and fall times, overshoots and undershoots
  - Clocks account for up to 50% of total power budget
  - Discrete frequency space, dynamic control schemes uncommon

→ What challenges these objectives?

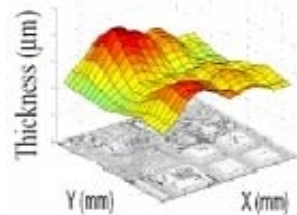


# Current Challenges

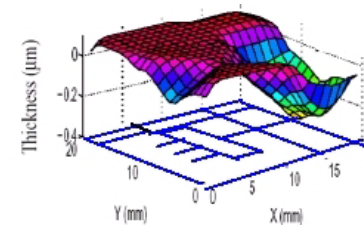
## Technology Scaling

- Increased uncertainty with process scaling
  - Process, voltage, temperature variations, noise, coupling
  - Affects design margin → over design, power & performance loss
- Increased power constraints
  - Increasing leakage, power (density, delivery) limitations

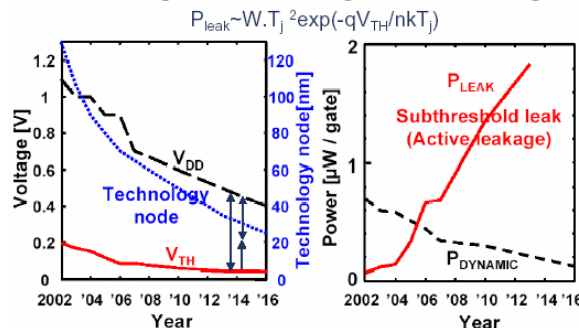
### Cu Thickness Variation



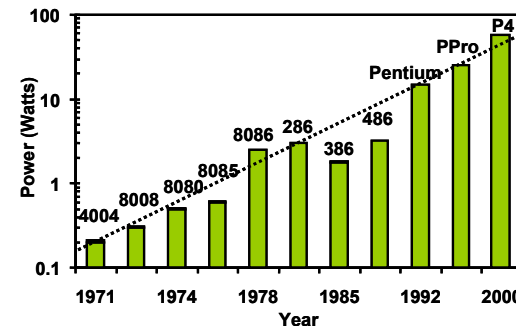
### ILD Thickness Variation



### Voltage/Leakage Scaling



### Power Scaling



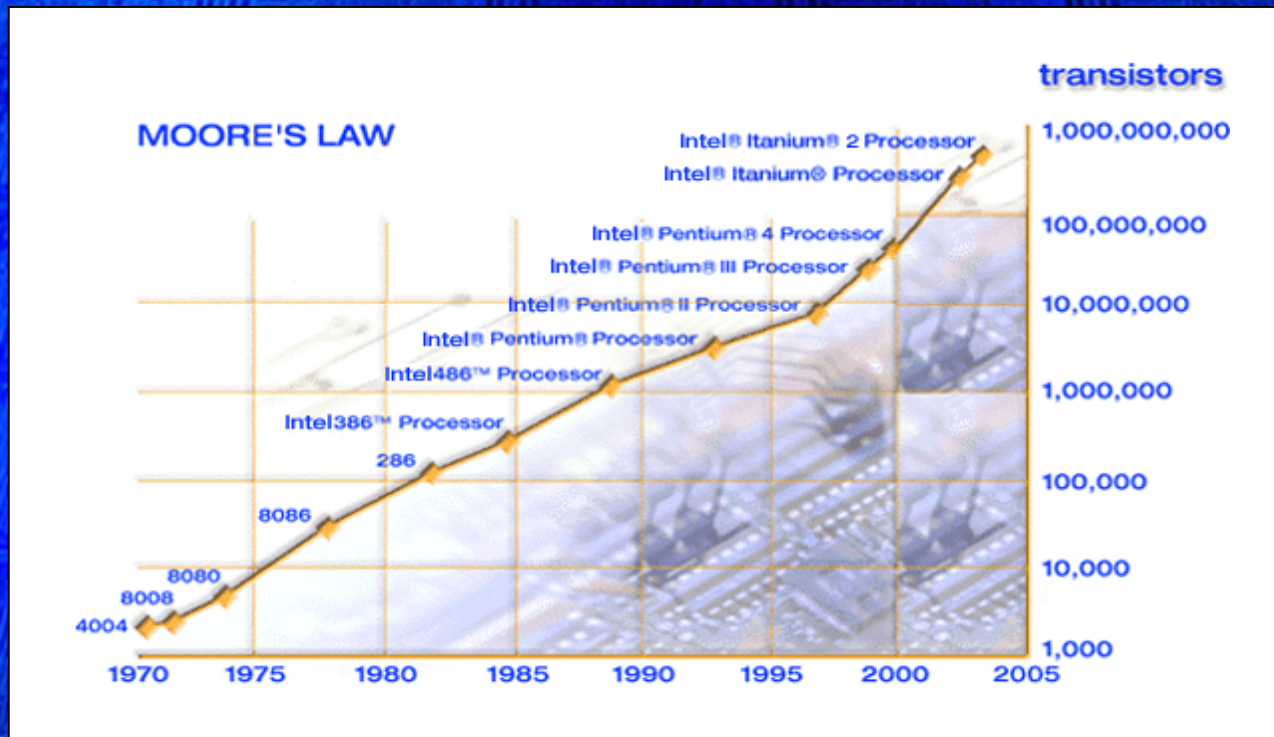
Source:  
INTEL  
Praegasus



# Current Challenges

## Transistor Count

- Moore's law increases logic density and "time-to-market" pressure
- More transistors mean:
  - Larger clock distribution networks
  - Higher capacitance (more load and parasitics)

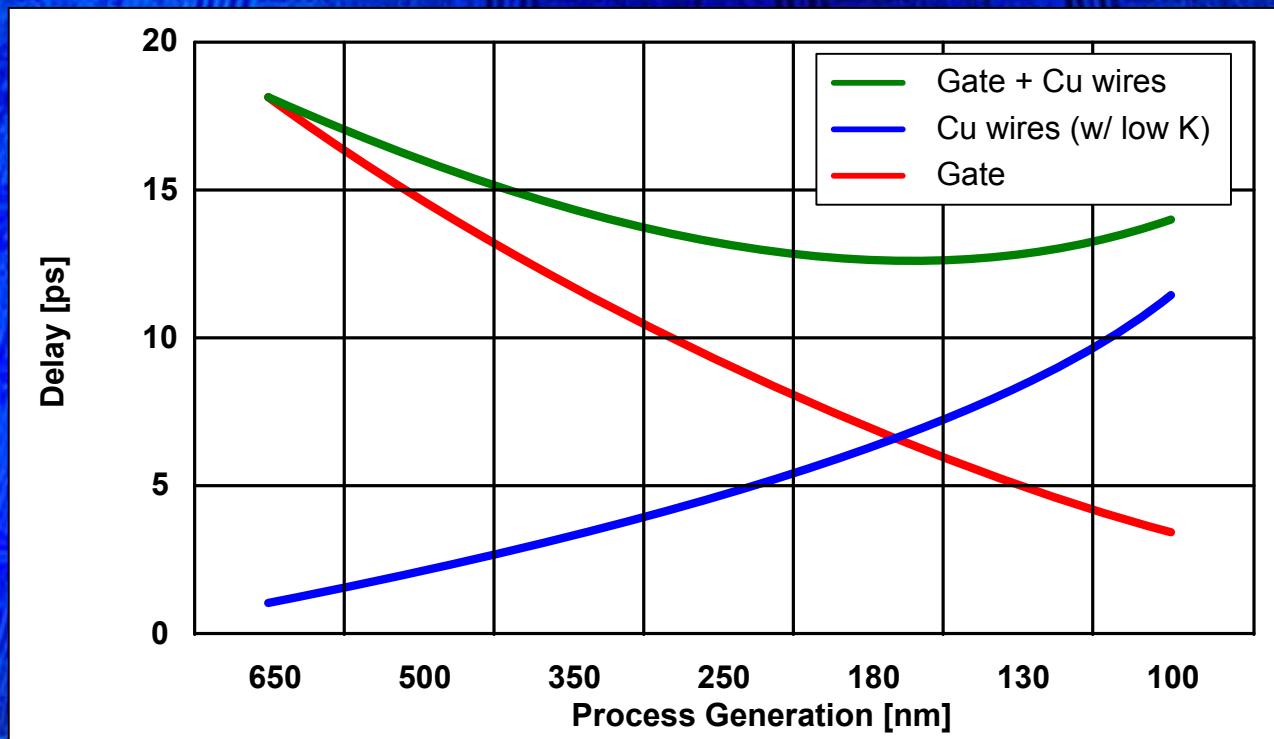




# Current Challenges

## Interconnect Delay

- With each new technology:
  - Gate delay decreases ~25%
  - Wire delay increases ~100%
  - Cross-chip communication increases
    - Clock needs multiple cycles to cover die

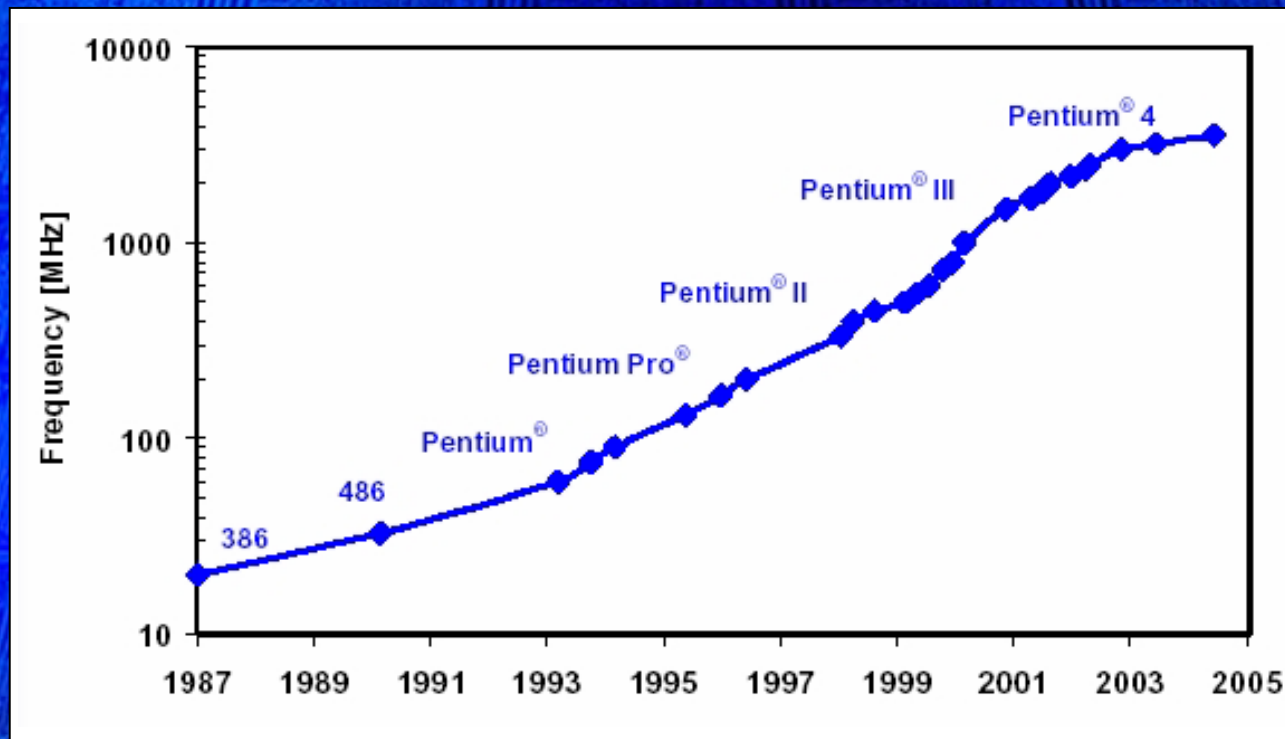




# Current Challenges

## Clock Frequency

- End of frequency paradigm
  - Power is linearly related to frequency with no voltage scaling
  - Power is cubically related to frequency and voltage scaling
  - Performance is not linearly related to frequency

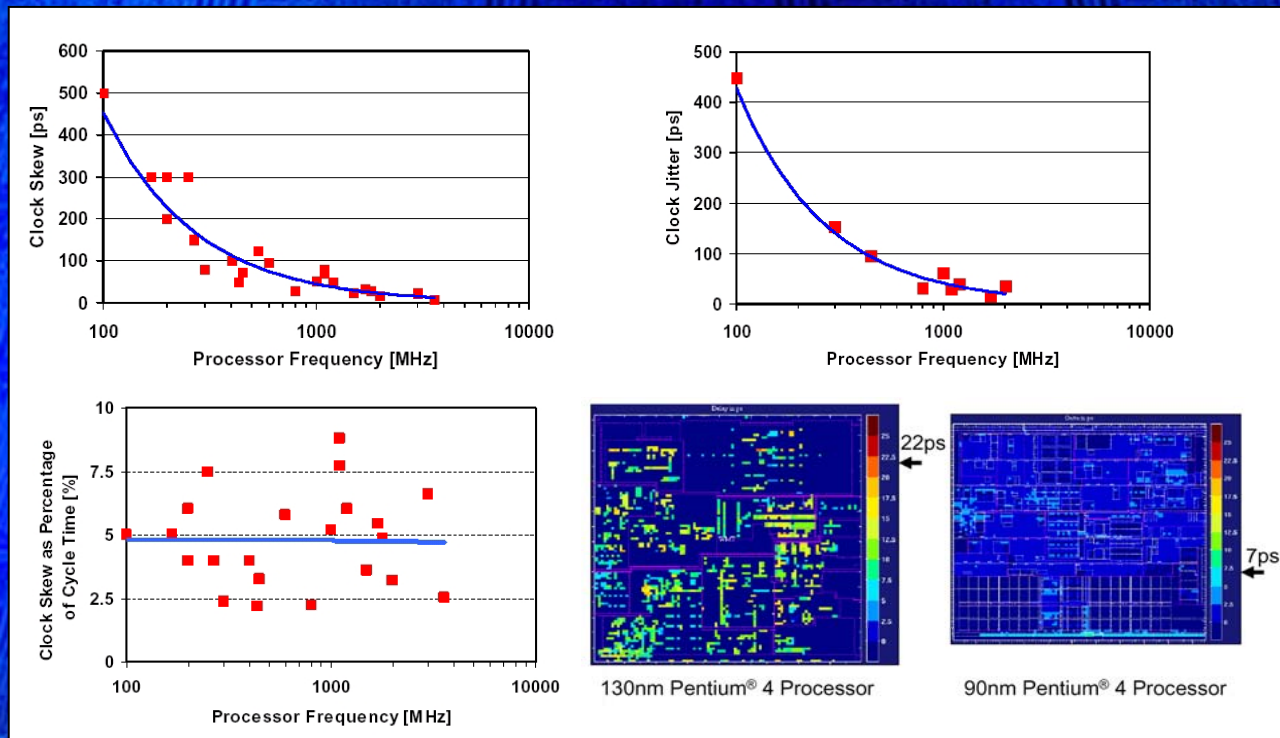




# Current Challenges

## Clock Skew and Jitter

- Can clock skew remains at ~5% of total clock budget?
  - 50ps at 1GHz, 5ps at 10GHz...
  - Additionally, setup and hold time increases
  - “Useful” cycle time margin decreases





# Current Challenges

## Summary

- Increased transistor density
  - More uncertainty with process scaling
- Larger microprocessors
  - 100 Million ~ 1 Billion transistors/chip
- Reduced power margins
  - Power-aware and power-constrained design necessary
- Very high speeds
  - Multiple GHz clock rates still required
- Shrinkage of useful cycle time and margins
  - Clock and skew targets harder to meet

→ Clock and power design becomes increasingly challenging



# Outline

- Current Challenges
- **Industry Examples**
- Future Directions
- Summary



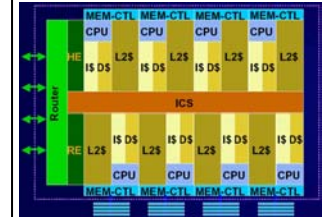
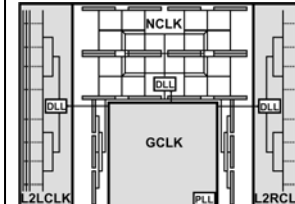
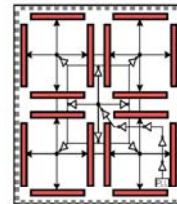
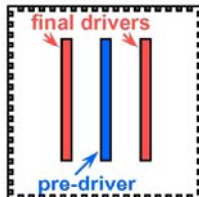
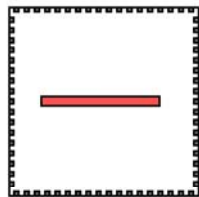
# Industry Examples

## SPSoC Microprocessors

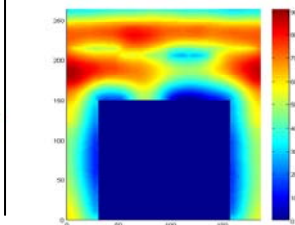
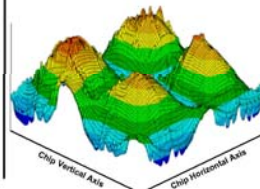
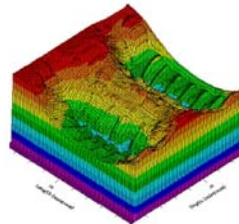
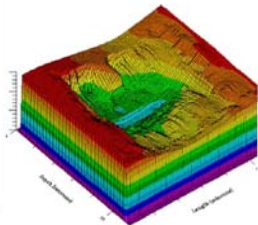
- DEC/Compaq Alpha

Product	21064	21164	21264	21364	Piranha
Frequency	166MHz	300MHz	600MHz	1.3GHz	1.25GHz
Transistors	1.7M	9.3M	9.3M	152M	N/A
Process	0.75um 4ML	0.5um, 4ML	0.35um, 6ML	180nm	N/A
Power	25W	50W	72W	155W	N/A

Clock Floorplan



Clock skew plot



N/A

Single Cores

Single Core SoC Multi-Core SoC



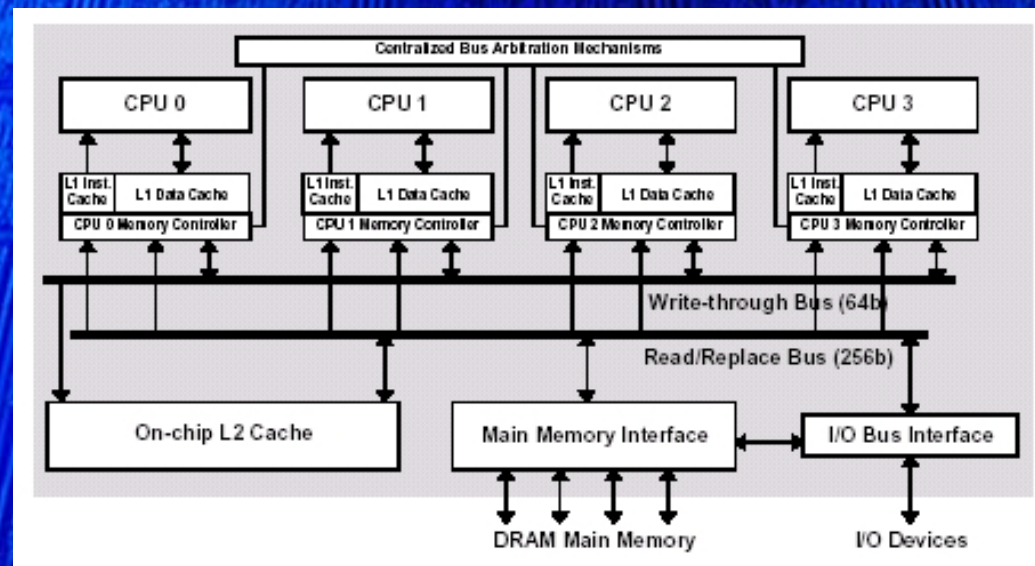
❖ **Trend:** more complex core to improve performance, more complex clocks (?)



# Industry Examples

## MPSoC Microprocessors

- Stanford Hydra CMP
  - 4 processors
  - Shared L2 cache
  - 2 internal buses



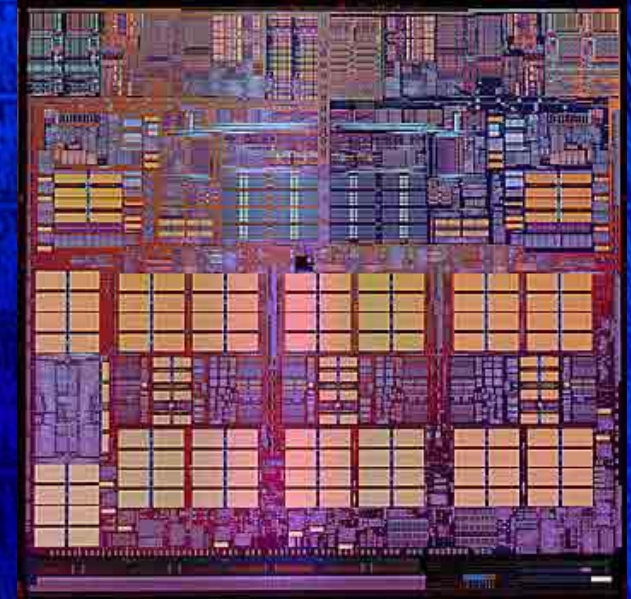
❖ **Trend:** multiple simple cores on die, bus communication, shared cache



# Industry Examples

## MPSoC Microprocessors

- IBM Power4
  - 2 cores
  - $F = 1.4\text{GHz}$
  - Single clock over entire die
    - Balanced H-tree driving global grid
    - Measured clock skew below 25ps
  - Power ~85W
    - When processing requires high throughput instead of single stream performance complexity, one core can be turned off
  - 180nm SOI process, 174M transistors



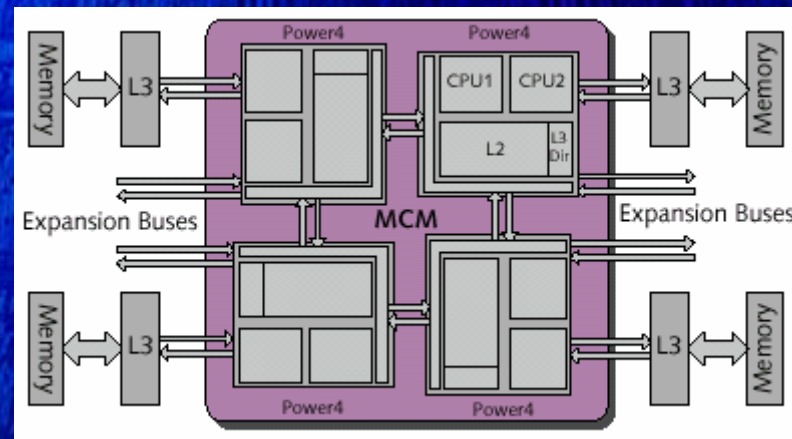
❖ **Trend:** multiple processors on die, bus communication, shared cache



# Industry Examples

## MPSoC Microprocessors

- IBM Power4
  - 4 POWER4 chips into single module (MCM)
  - The POWER4 chips connected via 4 128-bit buses
  - Up to 128MB L3 cache
  - Bus speed  $\frac{1}{2}$  processor speed
  - Total throughput  $\sim 35$  GB/s



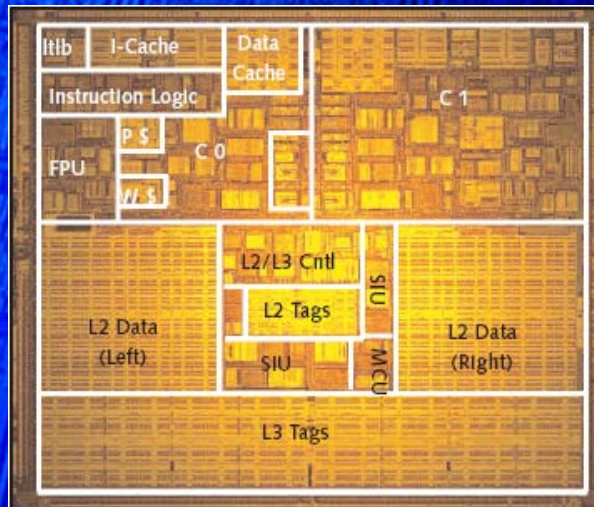
❖ **Trend:** multiple processors on MCM, on module bus communication, huge cache



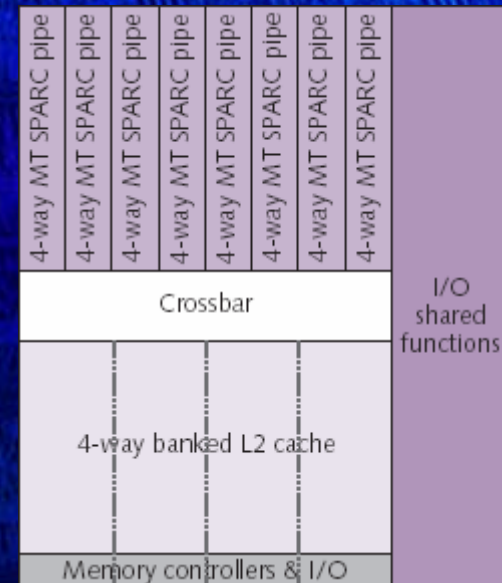
# Industry Examples

## MPSoC Microprocessors

- Sun UltraSparc IV+
  - 2 cores
  - Shared 2MB L2 cache
  - F = 1.8GHz
  - TI 90nm CMOS, 300M transistors



- Sun Niagara
  - 8 cores
  - 4 threads per core
  - Shared 3MB L2 cache



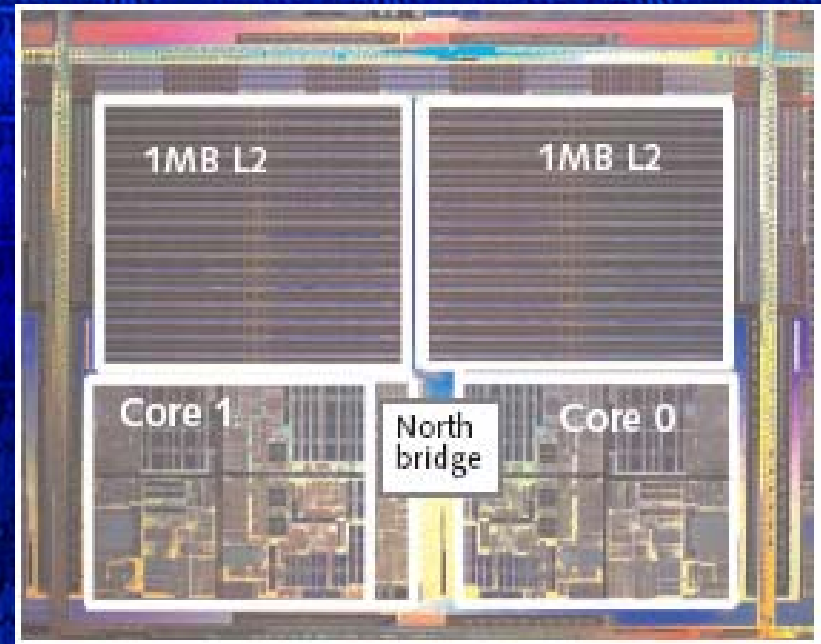
❖ **Trend:** multiple processors on die, bus communication, shared cache



# Industry Examples

## MPSoC Microprocessors

- AMD Dual Core Opteron
  - 2 Cores
  - F = 1.8GHz
  - 106M transistors
  - P = 70W
  - HyperTransport I/O
  - 2 1MB L2 caches
  - 220M transistors



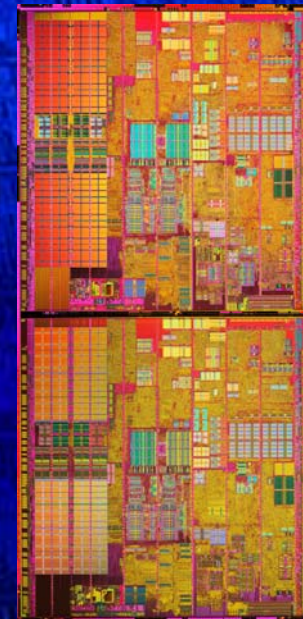
❖ **Trend:** multiple processors on die, bus communication, unshared cache



# Industry Examples

## MPSoC Microprocessors

- Intel® Pentium® D
  - Two Processors on MCM
  - 2 1MB L2 caches
  - 90nm CMOS
  - F = 3.2GHz
  - 230M transistors



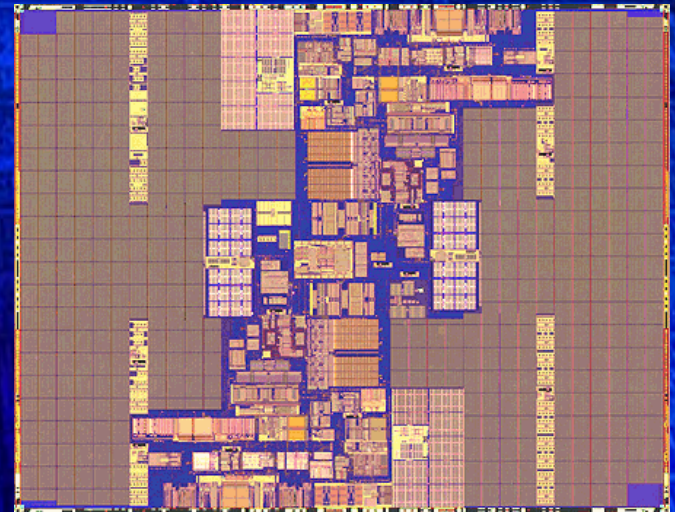
❖ **Trend:** multiple processors on MCM, unshared cache



# Industry Examples

## MPSoC Microprocessors

- Intel® Itanium® Montecito
  - 2 cores
  - F ~ 1.5GHz
  - P = 100W
  - 90nm CMOS, 1.72B transistors
  - 2 12MB L3 asynchronous caches
  - 596mm<sup>2</sup>
  - Multiple clock domains
  - Foxton power management circuitry
    - Dynamic voltage and frequency adjustment
    - Based on current and noise sensing



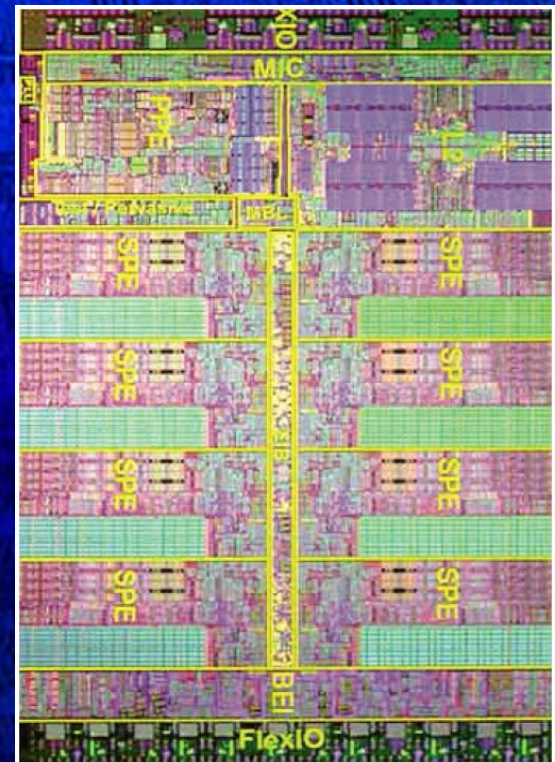
❖ **Trend:** multiple cores on die, bus communication, unshared cache



# Industry Examples

## MPSoC Microprocessors

- STI (Sony, Toshiba, IBM) Cell
  - 9 Cores / 10 Execution threads
  - $F = 4.6 \text{ GHz}$  @  $V = 1.3\text{v}$ 
    - Covers 85% of die
  - $P = 50 \sim 80\text{W}$  (estimated)
  - 512KB L2 Cache
  - Die Size:  $221 \text{ mm}^2$
  - 234M Transistors
  - 90nm SOI technology (Low K, 8 layers, Cu interconnect)
  - 6.4GT/s I/O interface
  - 4 x 128 bit internal bus (ring)



❖ **Trend:** multiple mixed processors on die, bus communication, shared cache



# Industry Examples

## Summary

- Microprocessor industry seems to lean towards similar MPSoC models:
  - Multiple leveraged processors on die
  - Bigger shared caches
  - On-die communication between processors
  - High-speed I/O links
- **Impact on clocks**
  - Global clocking solutions are not likely
    - Modularity is required
    - Die size is overwhelming
  - Multiple smaller, “simpler” clock domains
    - Easier local design, better skew and jitter control
    - Global communication is still a challenge
  - Intra-domain communication
    - Synchronous, asynchronous?



# Industry Examples

## Summary

- Microprocessor industry seems to lean towards similar MPSoC models:
  - Multiple leveraged processors on die
  - Bigger shared caches
  - On-die communication between processors
  - High-speed I/O links
- **Impact on power**
  - Power envelope is not increasing!
  - Multiple clock domains allow for more power control
    - More dynamic voltage/frequency scaling
    - More meticulous clock gating schemes
  - Power management becomes paramount
    - Needs to monitor temperature, core activity, load balancing...
  - Multiple power domains should be next...



# Industry Examples

## Summary

- **Increased clock and power complexity**
  - Clock distribution coverage area growth
    - More modular and efficient solutions required
  - Collapse of single-clock paradigm
    - Multiple clock domain partitioning advantage
    - Data transfer between domain complication
  - Static voltage and frequency approach fracture
    - Voltage and frequency scaling necessary to reduce power
    - Efficient power management and leakage control required
    - Dynamic clock generation scheme emergence
  - Integration of power management on die required
    - Multitude of sensors (temperature, voltage, activity...)
    - Control of power and frequency levels globally, regionally, possibly locally



# Outline

- Present Challenges
- Industry Examples
- **Future Directions**
- Summary



# Future Directions

## Clock Distribution

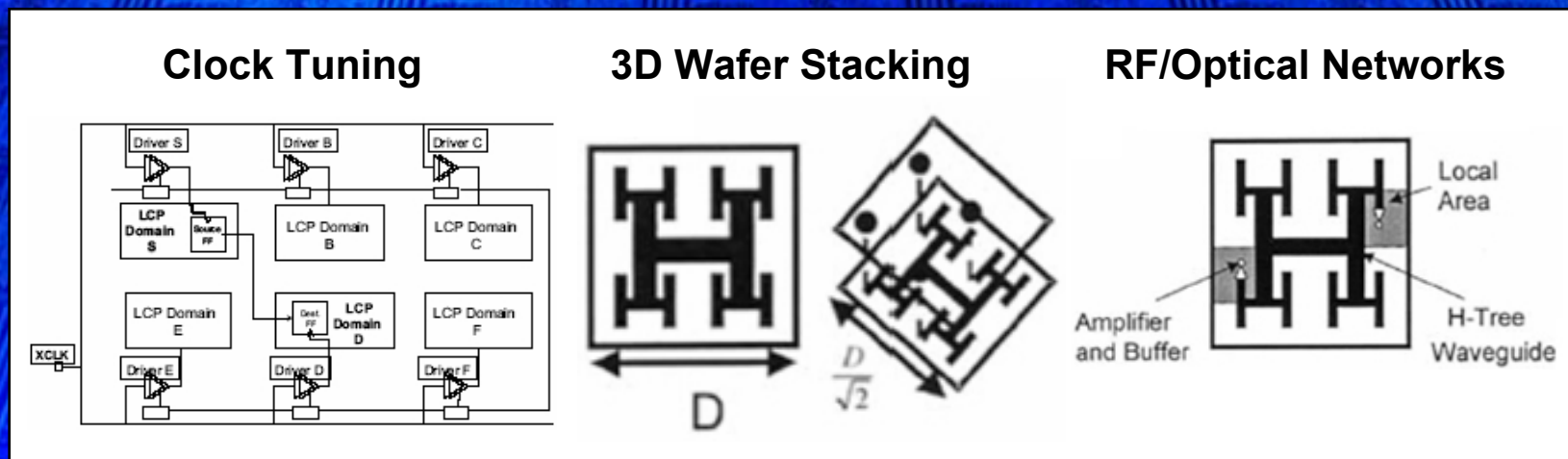
- Multiple clock domains
  - Low skew and jitter ALWAYS a must
  - Clock modeling requires more accuracy
    - Within-die variations, inductance, crosstalk, electromigration, self-heat, ...
  - Floor plan modularity
    - Think adding/removing cores seamlessly!
  - Hierarchical clock partitioning
    - Reduce global clock and possibly relax its requirements
    - Generate “locally”-used clock “locally”
    - Implement clock domain deskewing techniques
    - Bound clock problem into simple, reliable, efficient domains



# Future Directions

## Clock Distribution

- Multiple clock domains
  - Global clock interconnect is challenged
  - Exotic options are being investigated



- Data transfer between clock domains
  - Latency and determinism issues
  - GALS/GALS/GALA options?



# Future Directions

## Asynchronous Logic

- Advantages...
  - Performance
    - Potentially higher
    - Not limited by slowest component
  - Power
    - Potentially much lower
    - No clock power overhead
    - Inactive components consume “only” leakage power
    - Better EMI
  - Design Complexity
    - Easier circuit synthesis
    - Possibly more scalable (no timing issues)
    - Synchronization with clock domains requires no clock relationship (handshaking, flow control)



# Future Directions

## Asynchronous Logic

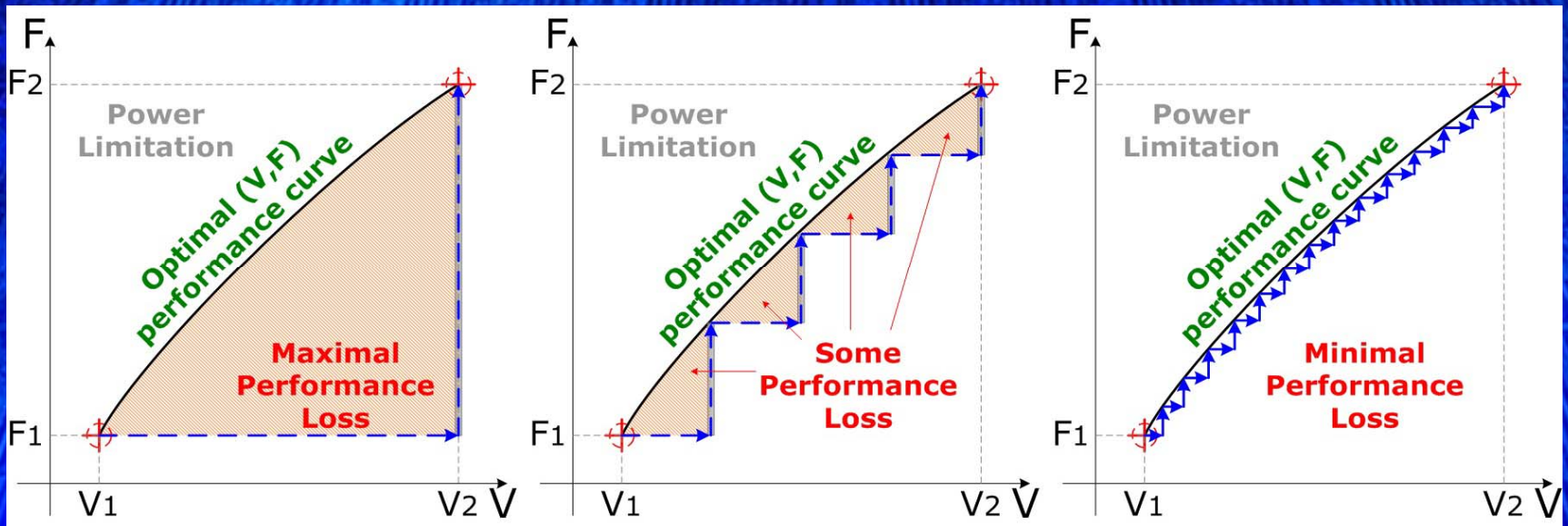
- Disadvantages
  - Performance
    - Area penalty for similar functionality
  - Power
    - Extra components consume power
  - Design Complexity
    - Potentially more complex circuit design
    - Vulnerable to circuit glitches
    - Test and debug more complex
    - Lack of CAD Tools
- Asynchronous products exist!
  - Embedded processors from Philips Semiconductors, Motorola...



# Future Directions

## Clock and Power Convergence

- Dynamic voltage and frequency scaling (DVS)
  - Graphical representation



Basic power management  
Static frequency scaling

Coarse power management  
Static frequency scaling

Optimal power management  
Dynamic frequency scaling



# Future Directions

## Clock and Power Convergence

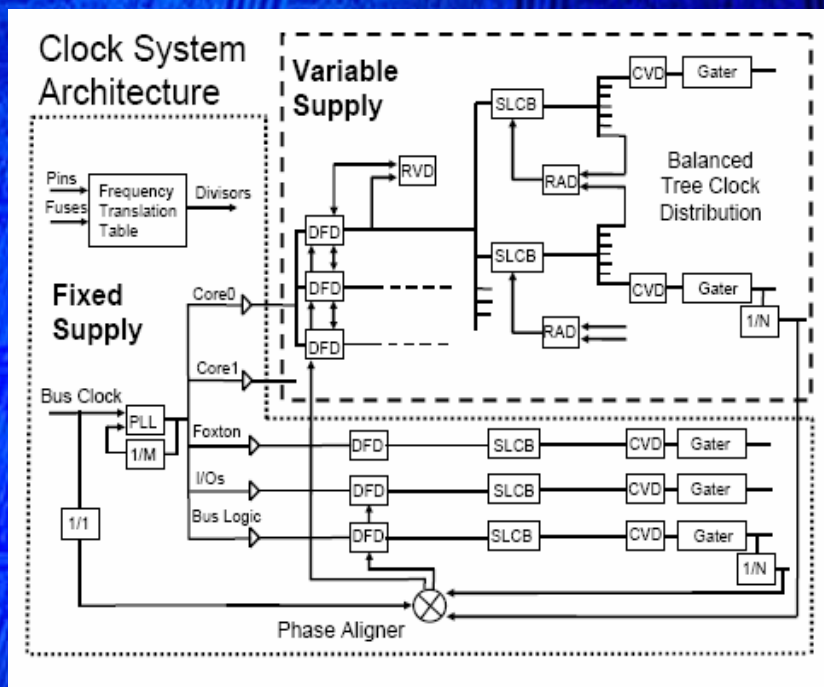
- Dynamic voltage and frequency scaling (DVS)
  - Dynamic frequency control
    - PLL-based schemes are not optimal for such tasks
    - More digital solutions start being proposed (Intel Itanium's Digital Frequency Divider)
  - Dynamic power management
    - Requires early micro-architectural estimation
      - Power state definition (on, idle, off, ...)
    - System complexity increases with number of parameters
      - Number of cores, threads, ...
    - Additional on-die logic blocks
      - Microcontroller
      - Sensors of all kinds
    - Close interaction with clock system
      - Powering off functional units, cores, ...
      - Slowing down functional units, cores, ...



# Future Directions

## Clock and Power Convergence

- Intel® Itanium® Montecito - Clock system architecture
  - Each core split into 3 clock domains on variable power supply
  - Each domain controlled by Digital Frequency Divider (DFD) generating low-skew variable-frequency clocks; fed by central PLL and aligned through phase detectors



- Regional Voltage Detector (RVD): supply voltage monitor
- Second level clock buffer (SLCB): digitally controlled delay buffer for active deskewing
- Regional Active Deskew (RAD): phase comparators monitoring and adjusting delay difference between SLCBs
- Clock Vernier Device (CVD): digitally controlled delay buffer



# Future Directions

## Clock and Power Convergence

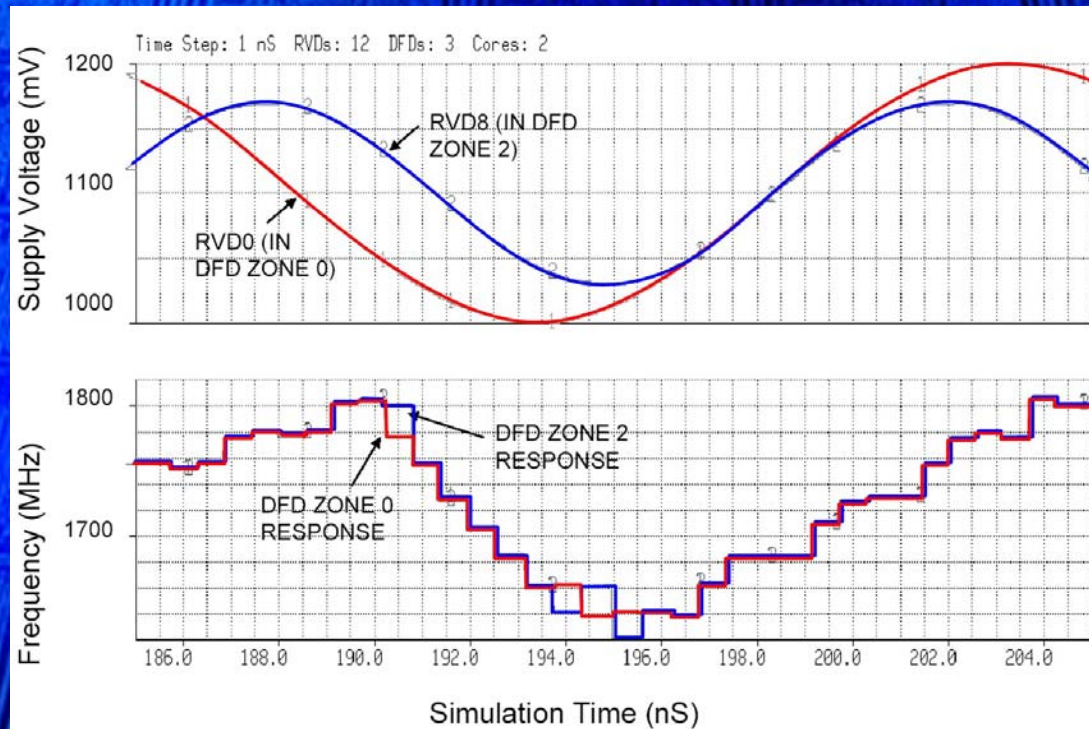
- Intel® Itanium® Montecito - Power management (Foxton)
  - Dynamic voltage-scaling power management system
  - 4 on-die sensors
  - On-die microcontroller
    - Power and temperature measurement
    - Voltage and frequency modulation
    - 8 $\mu$ s power/temperature sampling interval
  - Embedded firmware
    - Power, temperature, or calibration measurements
    - Power: closed-loop power control and system stability check
    - Temperature: thermal sensor readout (junction temperature below 90°C monitoring) and power-control communication
    - Calibration: power-measurement accuracy check



# Future Directions

## Clock and Power Convergence

- Intel® Itanium® Montecito - Noise management
  - Voltage to Frequency Converter (VFC): dynamic core frequency adjustment as a function of voltage
    - VFC locks onto and tracks local supply voltage with RVD
    - 4 RVDs per DFD controlling VFC





# Summary

- Clock and power designs trends
  - Increased transistor density, larger microprocessors
  - GHz+ frequencies, shrinkage of useful cycle time and margins
  - High-quality clock design ALWAYS aimed towards low skew, low jitter, low power
- Convergence of clock and power designs
  - Multiple clock domains (synchronous or not)
    - Hierarchical clock complexity
  - Dynamic voltage and frequency scaling
    - Dynamic frequency generation systems
    - Dynamic power management systems
    - Close interaction for maximum performance



# Acknowledgments

Thanks to all my colleagues within Intel and the engineer community at large for providing the technical innovations contained in the several microprocessors I discussed today