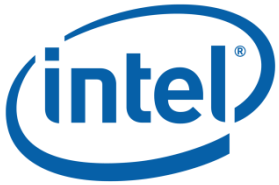
MPSoC '07
25-29 June, Japan



Clock System Design Risks and Opportunities under the MPSoC Paradigm

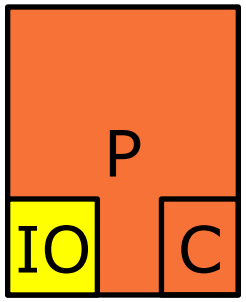
Olivier Franza
Olivier.Franza@intel.com

Outline

- Monolithic Microprocessor Clock System Designs
- MPSoC Needs – the "Building-block" Methodology
- Future MPSoC Clock System Designs
- Summary

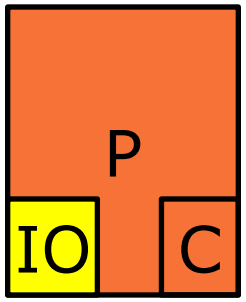
Outline

- Monolithic Microprocessor Clock System Designs
- MPSoC Needs – the "Building-block" Methodology
- Future MPSoC Clock System Designs
- Summary



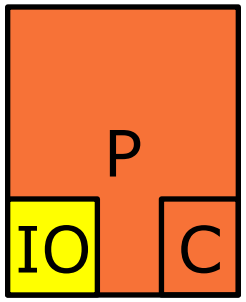
Monolithic Microprocessor Clock System Designs

- Requirements
  - Single main clock frequency
    - Big clock distribution
    - Single clock generation block
  - Divided or derived clocks for IO
  - Simple boundary crossings



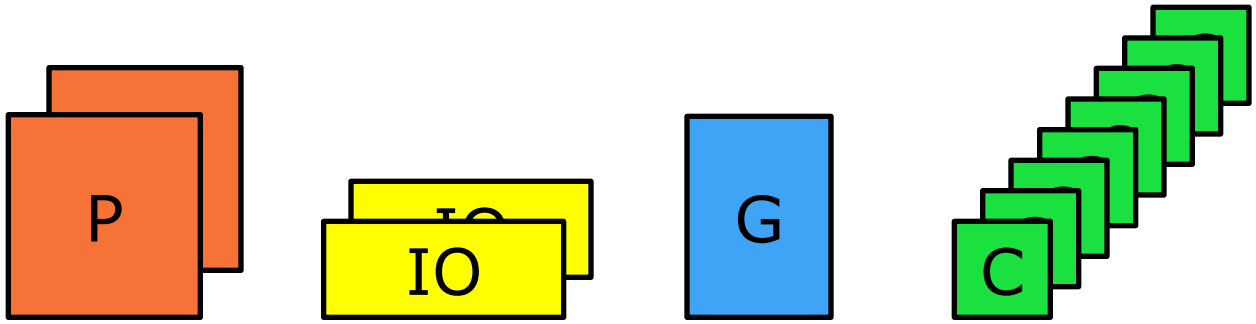
Monolithic Microprocessor Clock System Designs

- Advantages
  - Simple design
- Drawbacks
  - Not optimized for power – single frequency operating point
  - Not modular – design targeted to specific product



Outline

- Monolithic Microprocessor Clock System Designs
- MPSoC Needs – the "Building-block" Methodology
- Future MPSoC Clock System Designs
- Summary



MPSoC Needs – the "Building-block" Methodology

- Where does performance increase reside for next-generation microprocessors?
  - Parallelism: multi-cores, many-cores, multi-threading, virtualization
    - Numerous nimbler (heterogeneous) processing units
    - Added application-specific hardware (graphic, chipset, …)
    - Trade frequency for parallelism
      - It's a better choice for performance, power consumption, design complexity, and turn-around-time
      - No more frequency boosts? Still good for parts binning and marginal product life cycle updates
  - Larger more efficient on-chip caches
  - Increased bandwidth

MPSoC Needs – the "Building-block" Methodology

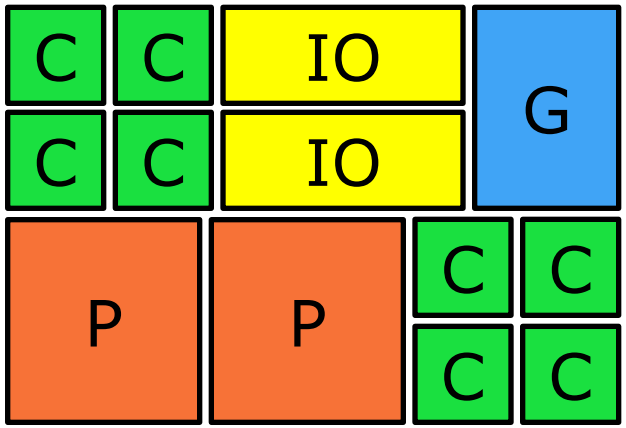
- "Building-block" requirements
  - Modularity
    - Ease module placement by defining standard block size (Lego)
    - Allow multiple interface attachments
    - Define modular network between modules with common protocol
  - Interchangeability
    - Application specific blocks can be replaced depending on product target
  - Manageability
    - Each module is treated as black box, fully verified and validated
    - Interfaces designed within modules ensuring correct-by-assembly design
    - Top-level integration only focuses on connectivity and high level system design

MPSoC Needs – the "Building-block" Methodology

- "Building-block" requirements
  - Compatibility
    - Blocks should be designed to optimize die process performance
    - Incompatible modules should be placed on different dies, MCM or 3D-stack
  - Interoperability
    - Common interface for all modules
    - Common protocol for module communication
    - Implies overhead but simplifies communication
  - Self-sufficiency
    - Blocks should not depend on other blocks to operate
    - Implies own global circuits (power, clocks, test…)

Outline

- Monolithic Microprocessor Clock System Designs
- MPSoC Needs – the "Building-block" Methodology
- Future MPSoC Clock System Designs
- Summary



Future MPSoC Clock System Designs

- "Building-block" requirements
  - Modularity
    - Clock modularity calls for self-contained module-level clock domains
    - Full-die clock distributions undermine modularity
    - Clock generation integrated into modules
    - Common protocol must be clock aware (GSLS, GALS)
  - Interchangeability
    - Application specific blocks can have different clock requirements
  - Manageability
    - As black box, clock system must be fully operational at the module level (reference input clock pins, specific power supply for clock generation, frequency control…)

Future MPSoC Clock System Designs

- "Building-block" requirements
  - Compatibility
    - Incompatible modules should be placed on different dies
    - 3D global clock distributions?
  - Interoperability
    - Common interface & protocol for module communication
    - Common clock domain crossing design within each module
  - Self-sufficiency
    - Blocks should not depend on other blocks to operate
    - Clock generation and clock domain crossing logic included in module
    - Modules should be optimized for power and frequency depending on workload within the whole die

Future MPSoC Clock System Designs

- Module optimization
  - DVFS (digital voltage & frequency scaling)
    - Technology trend for reduced operating range – Vmin and Vmax converge
      - Vmax decreases due to process technology scaling
      - Vmin increases due to reliability issues: dI/dt management is critical to achieve Vmin functionality
    - Voltage changes are slow
      - PLL-based clock generation lock times minor
      - Placing voltage regulators on-die would improve slew rate
    - DVFS → DFS
      - Need for high-frequency digital frequency generation as number of modules on chip increases
      - Clock chopping, frequency shifts reduce dI/dt impacts

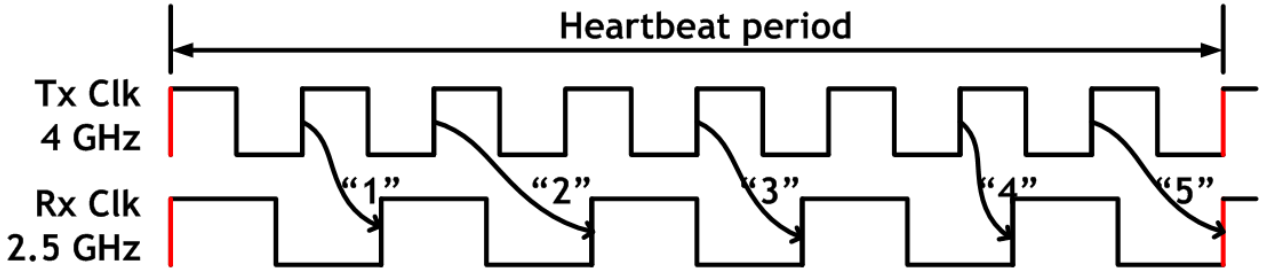
Future MPSoC Clock System Designs

- Common protocol for module communication
  - Microprocessors call for determinism
    - Post-Si test/debug
    - Lockstep operation
  - Performance impact due to increased domain crossing latency
    - Communication scheme required achieving minimal latency
  - Under a common reference clock, fractional clock data transfer (FCDT[1]) provides an elegant deterministic solution
    - Support for fine-grain DFS
    - Non-integral ratio clock frequencies
    - Not necessarily multiples of each other (5:4, 11:7)

1 Balasubramanian, S. Natarajan, N. Franza, O. Gianos, C., "Deterministic low-latency data transfer across non-integral ratio clock domains" – VLSI Design, 2006, 19th International Conference on, 3-7 Jan. 2006

Future MPSoC Clock System Designs

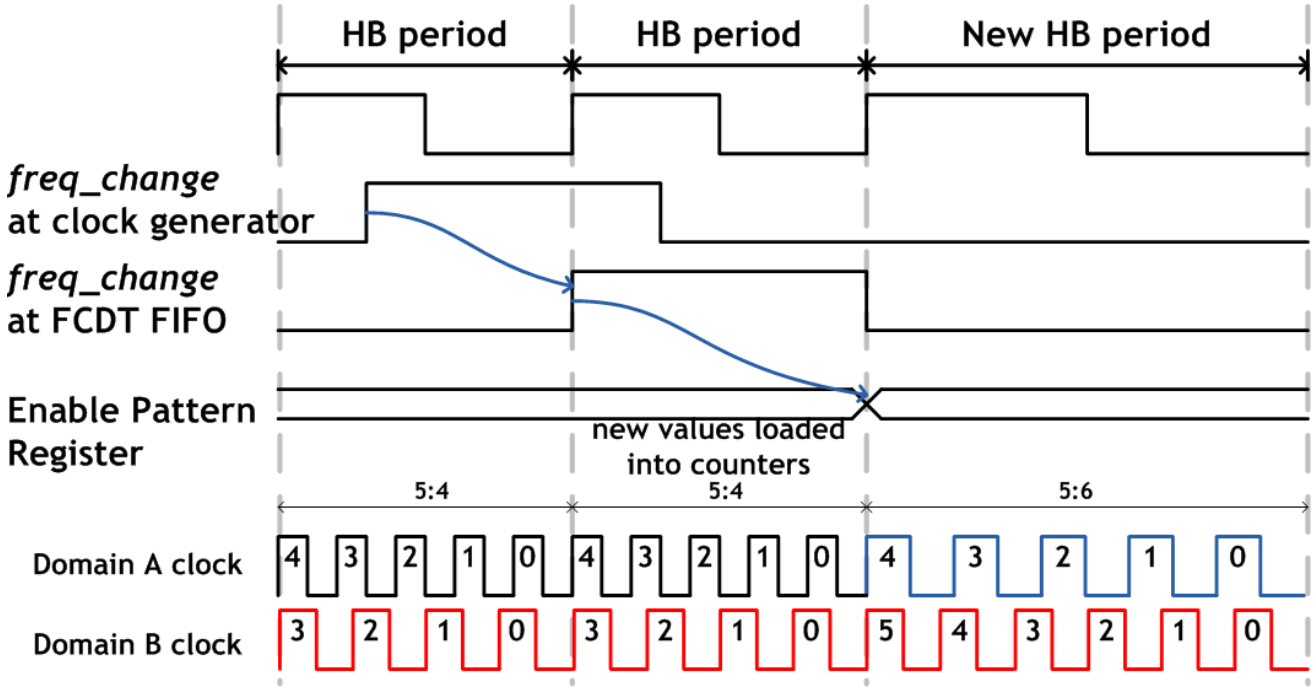
- FCDT[1] Overview
  - Consistently lower latencies than synchronizer-based approach
  - Fully deterministic scheme
  - Rational clock domain frequency only
  - Deterministic D(V)FS scheme support
  - Minimal logic overhead
  - Good scaling properties with FIFO size
  - Transmit and receive at pre-determined Tx and Rx clock edges
  - Enabled edges determined a priori by edge-choice algorithm



1 Balasubramanian, S. Natarajan, N. Franza, O. Gianos, C., "Deterministic low-latency data transfer across non-integral ratio clock domains" – VLSI Design, 2006, 19th International Conference on, 3-7 Jan. 2006

Future MPSoC Clock System Designs

- FCDT[1] & D(V)FS
  - Change frequency dynamically to optimize performance
  - Fine grain frequency changes (non-integral frequency ratios)
  - Non-stop data transfer during deterministic DFS events



1 Balasubramanian, S. Natarajan, N. Franza, O. Gianos, C., "Deterministic low-latency data transfer across non-integral ratio clock domains" – VLSI Design, 2006, 19th International Conference on, 3-7 Jan. 2006

Outline

- Monolithic Microprocessor Clock System Designs
- MPSoC Needs – the "Building-block" Methodology
- Future MPSoC Clock System Designs
- Summary

Summary

- "Building-block" approach is modifying the way clock systems are designed
- End of large single monolithic clock domains
- Future clock systems are made of:
  - Module-specific clock distributions
  - Well-defined low-latency deterministic clock domain crossing boundaries with common protocol between modules
  - Full D(V)FS support


intel®
Leap ahead™