A Low Cost High-Speed Source-Synchronous Interconnection Technique for GALS Chip Multiprocessor

Anh T. Tran, Dean N. Truong, Bevan M. Baas

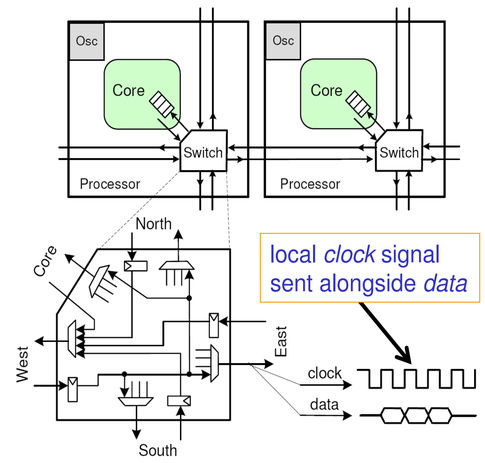
VLSI Computation Lab
University of California, Davis

Source-Synchronous Comm.

- An effective method for communication between clock domains
- A locally generated clock signal is used to latch the data into a receiving register or a dual-clock FIFO
- Requires *clock*, *data*, *valid*, and *request* signals for reliable transfer
- Supports a peak transfer rate of one data word per cycle

Source-Synch. Comm. for GALS

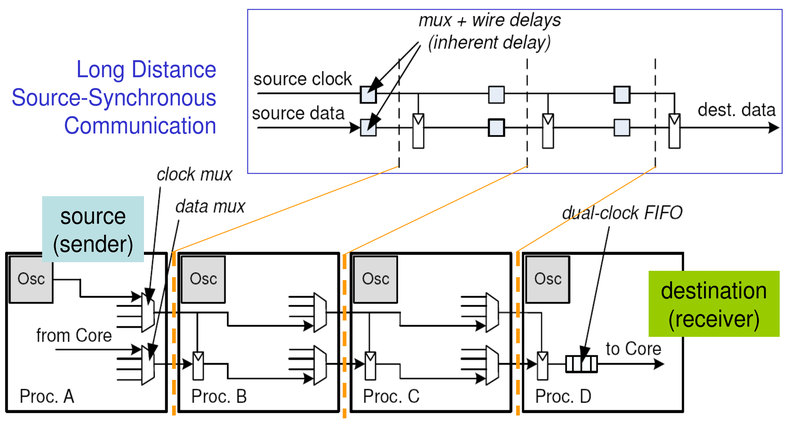
- GALS = **G**lobally **A**synchronous **L**ocally **S**ynchronous
- Each processor has its own oscillator (locally synchronous)
- This locally generated clock can be reused as the sender's clock



Circuit-switched Multi-clock Domain Network

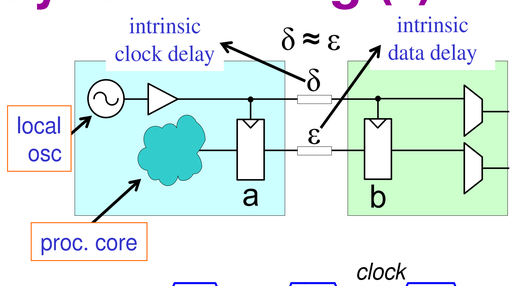
Circuit-switched Communication

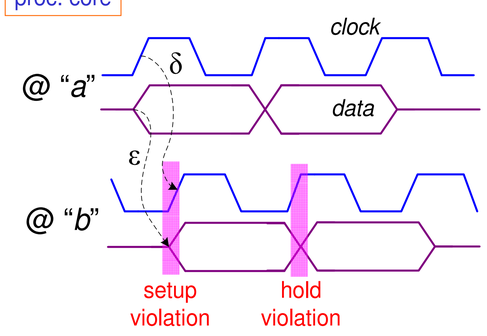
- Configurable muxes determine routing paths
  - Architecture is capable of communication between any two cores in a many-core array



Source-Synch. Timing (I)

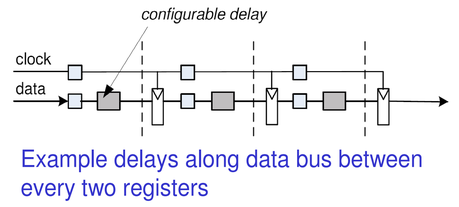
- Assume minimal to zero skew between source data and source clock…
  - can easily have setup and hold time violations
- Neither clock edge near each data word can be used to latch it successfully



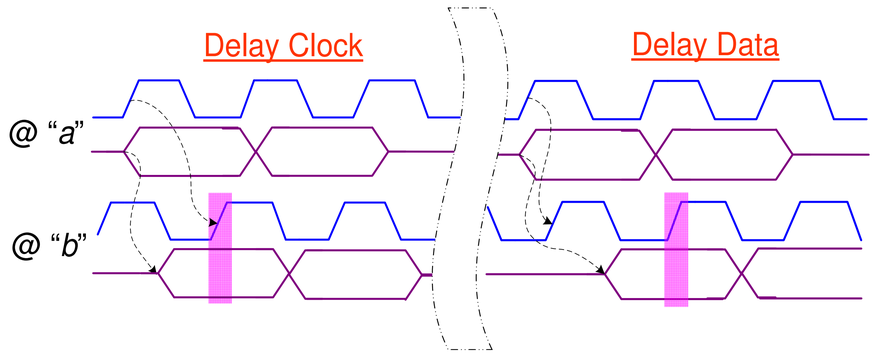


Source-Synch. Timing (II)

- Two potential solutions: add a delay to clock or add a delay to data:



Example delays along data bus between every two registers

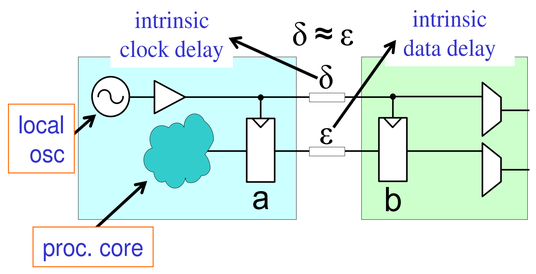


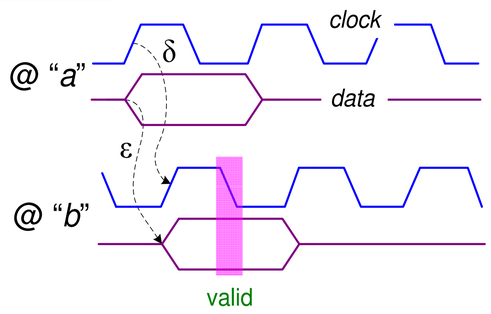
Timing Equations

- Constraints for clock delay (DLY_C):
  - $DLY_C - t_{clk\text{-}q} + t_{hold} < T$
  - $t_{setup} < DLY_C - t_{clk\text{-}q}$
- Constraints for data delay (DLY_D):
  - $DLY_D + t_{clk\text{-}q} + t_{setup} < T$
  - $t_{hold} < DLY_D + t_{clk\text{-}q}$

Alternating Edge-Triggered Timing

- Again, assume minimal to no skew between source data and source clock…
  - but now trigger on the opposite clock edge at destination
- Continue to use alternating clock edges for every other register



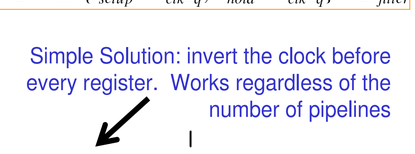


Alternating Edge Constraints

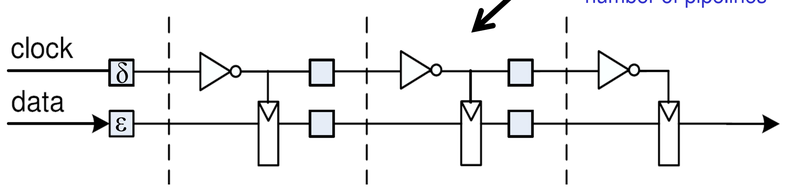
- Essentially, alternating the edges cuts the period in half and requires:
  1) $t_{clk\text{-}q} + t_{setup} < T/2$
  2) $T/2 + t_{hold} < T + t_{clk\text{-}q}$

  → $T > 2 \cdot max\{t_{setup} + t_{clk-q},\ t_{hold} - t_{clk-q}\} + 2t_{jitter}$
- We ignore the skew caused by inversion

Every register alternates its triggering-edge accordingly



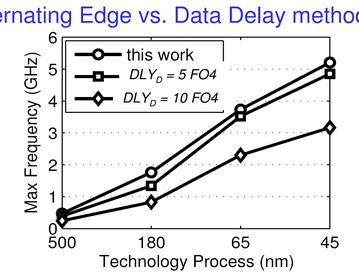
Simple Solution: invert the clock before every register. Works regardless of the number of pipelines

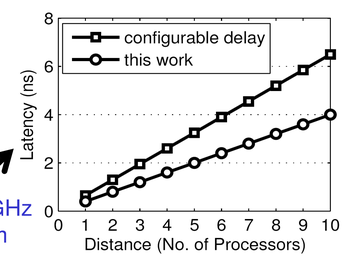


Ideal Max. Clock Freq. Analysis

- If we ignore realistic skew between clock and the data signals, then the theoretical limit of the clock frequency/period, is governed by the properties of the registers
- For standard cell designs the master-slave D flip-flop is commonly used for its robustness
  - However, its performance is sacrificed for robustness

Alternating Edge vs. Data Delay method





Conclusion & Future Work

- A cost efficient and robust source-synchronous architecture is presented that is compatible with GALS many-core arrays
  - Does not need static or adaptive circuits to readjust clock and/or data signals to meet timing
    - No configurable/adaptive delay elements or DLLs, PLLs & CDRs
    - It can achieve > 50% better maximum operating frequency and latency than the Data Delay method
- Future work:
  - Evaluate CAD auto place-route tool's ability to limit data-clock skew in deep submicron technologies
  - Variations of skew and jitter along the links
  - High speed serial data transmission over "harder to meet timing" parallel data bus transmission
  - Clock duty cycle and data signal distortion due to wire buffer propagation delay ($t_{pL\to H}$, $t_{pH\to L}$) mismatches

Acknowledgements

- NSF Grant 430090 and CAREER award 546907
- SRC GRC Grant 1598 and CSR Grant 1659
- Intellasys
- UC Micro
- Intel
- ST Microelectronics
- SEM
- J.-P. Schoellkopf, P. Cogez, Y.-P. Cheng, A. Gatherer, R. Krishnamurthy, K. Bowman, and M. Anders