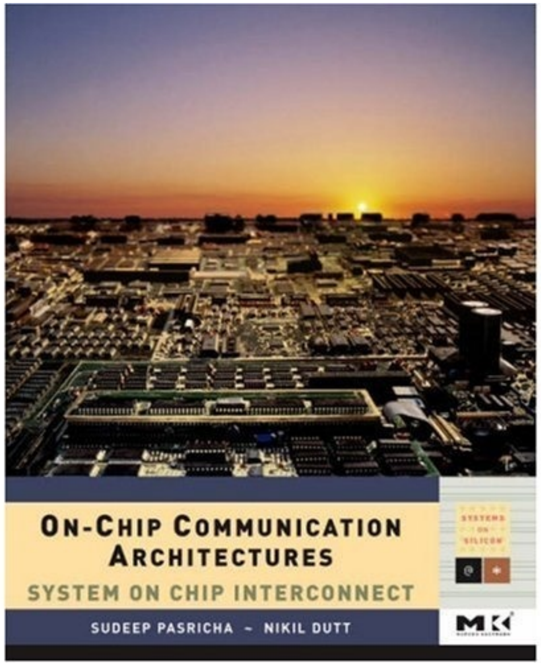
On-Chip Communication Architectures

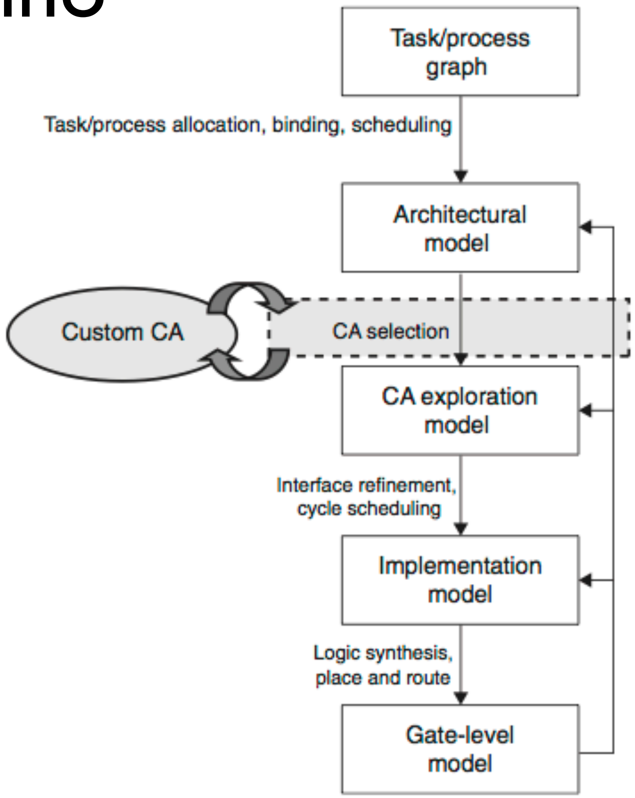

On-Chip Communication Architectures
System on Chip Interconnect
Sudeep Pasricha - Nikil Dutt
MK

Custom Bus-Based On-Chip Communication Architecture Design

ICS 295
Sudeep Pasricha and Nikil Dutt
Figures book chapter 8

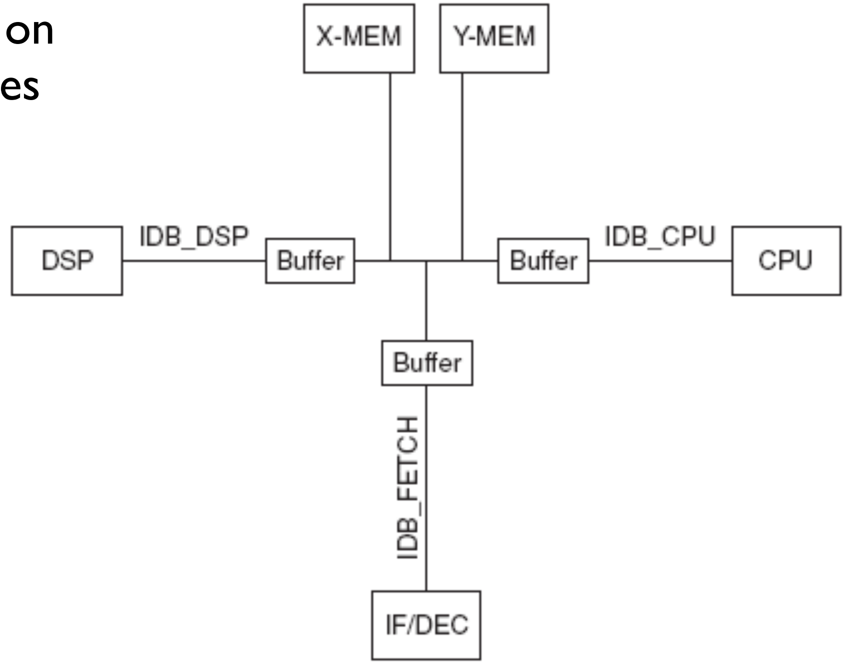
ESL Flow

- Seleção de arquitetura de comunicação dedicada
- Otimizar desempenho e potência


Task/process graph
Task/process allocation, binding, scheduling
Architectural model
Custom CA
CA selection
CA exploration model
Interface refinement, cycle scheduling
Implementation model
Logic synthesis, place and route
Gate-level model

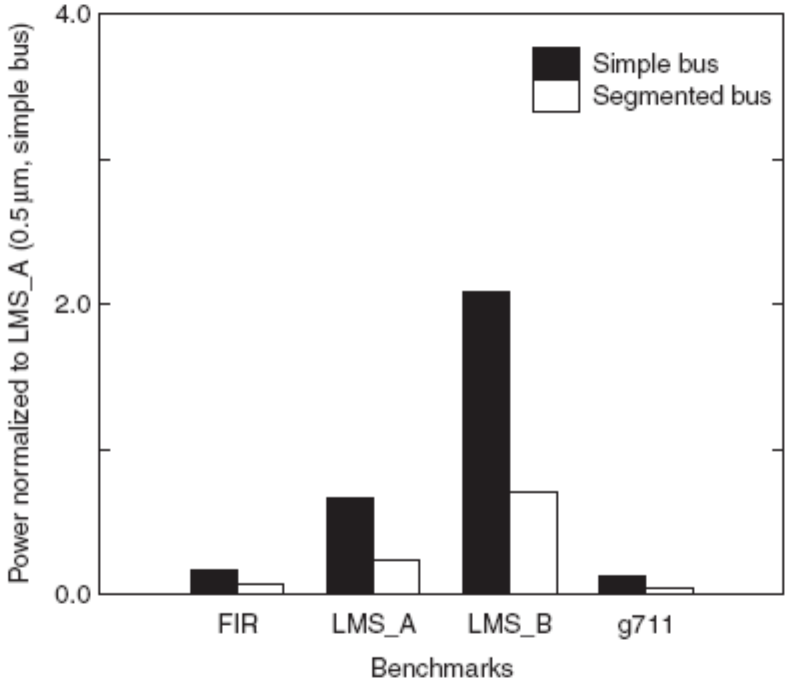
Split bus

- Split shared bus into multiple segments
- Split buses allow selective shutdown of unused bus segments, potentially saving energy
- Segmentation increases the parallelism by permitting parallel data transfers on different segments, which improves performance



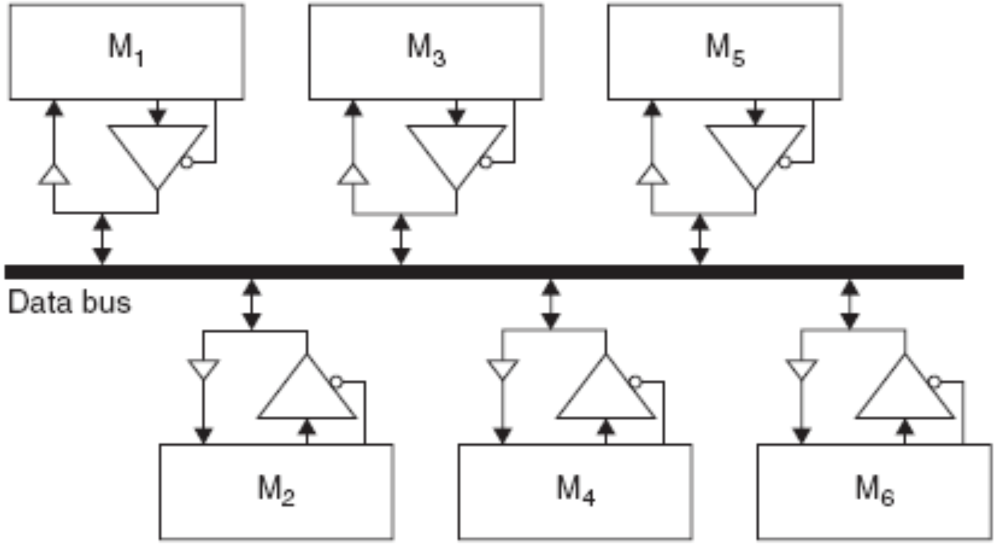
Split bus

- Total bus power consumption between segmented bus and shared bus architectures



Monolithic single shared bus architecture

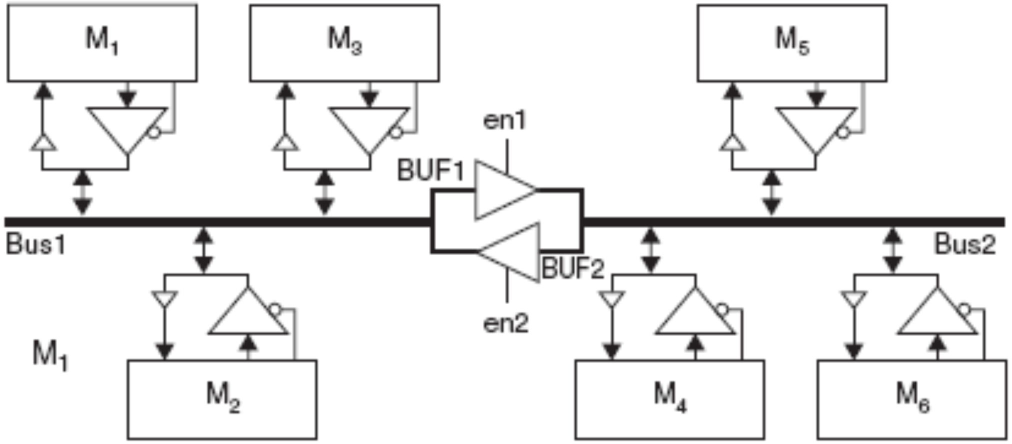
- Long propagation time
- Large capacitances



© 2008 Sudeep Pasricha & Nikil Dutt

Split bus architecture

- When *en1* is high, data can be transmitted from *bus1* to *bus2*, and when *en2* is high, data can be transmitted from *bus2* to *bus1*.
- When both *en1* and *en2* are low, the buses are isolated from each other.



Energy saving in split bus

- The components having the highest probabilities of data transfer should be kept on the same segment, so that only that segment of the bus architecture is active during the transfer, which saves energy

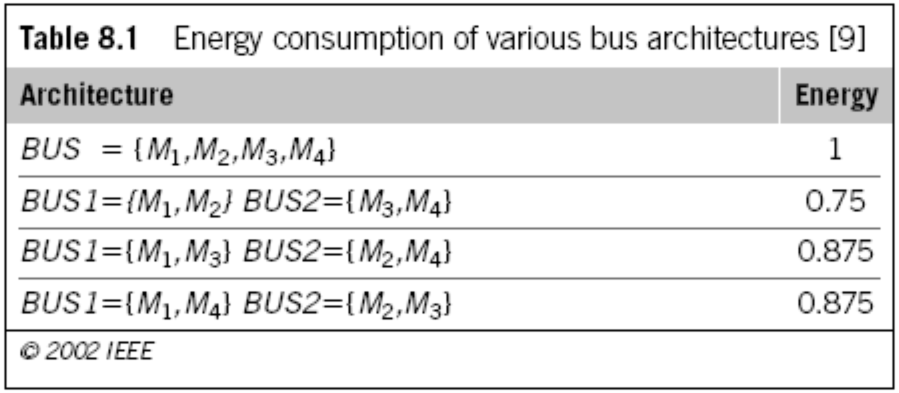
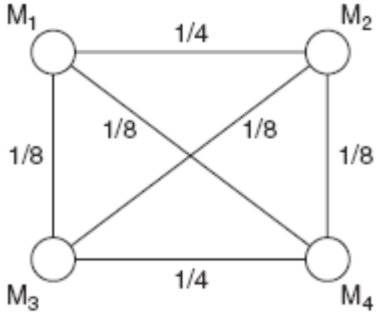


Table 8.1 Energy consumption of various bus architectures [9]

Architecture	Energy
$BUS = \{M_1,M_2,M_3,M_4\}$	1
$BUS1=\{M_1,M_2\}$ $BUS2=\{M_3,M_4\}$	0.75
$BUS1=\{M_1,M_3\}$ $BUS2=\{M_2,M_4\}$	0.875
$BUS1=\{M_1,M_4\}$ $BUS2=\{M_2,M_3\}$	0.875

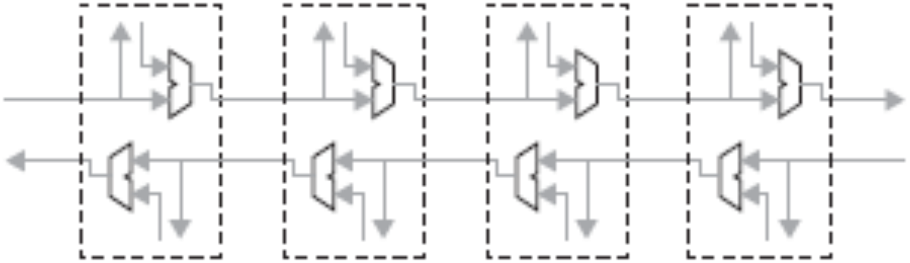
© 2002 IEEE

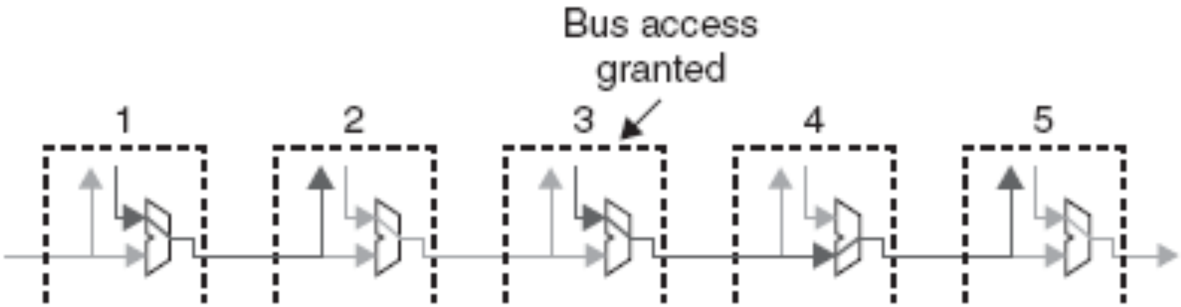


SAMBA (single arbitration, multiple bus accesses)

- allows multiple masters to access the bus with only a single bus arbitration grant.
- improve bus bandwidth and latency response

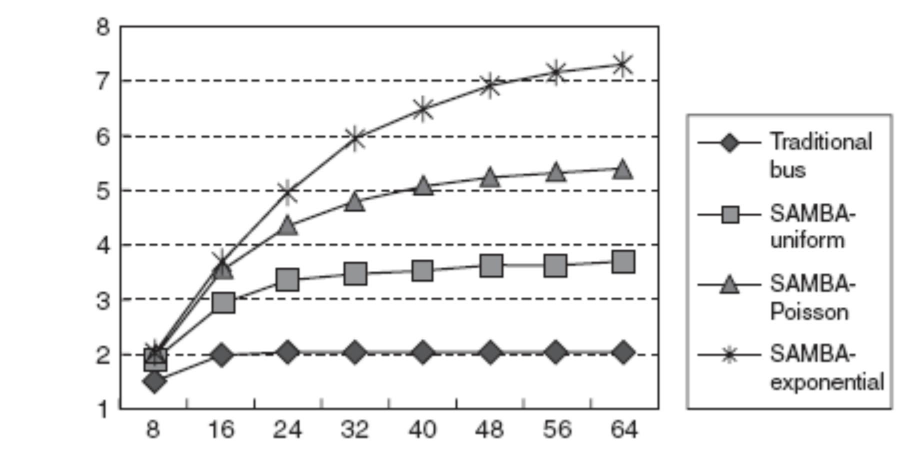
two separate buses, each of which is used for data transfer in a forward or backward direction



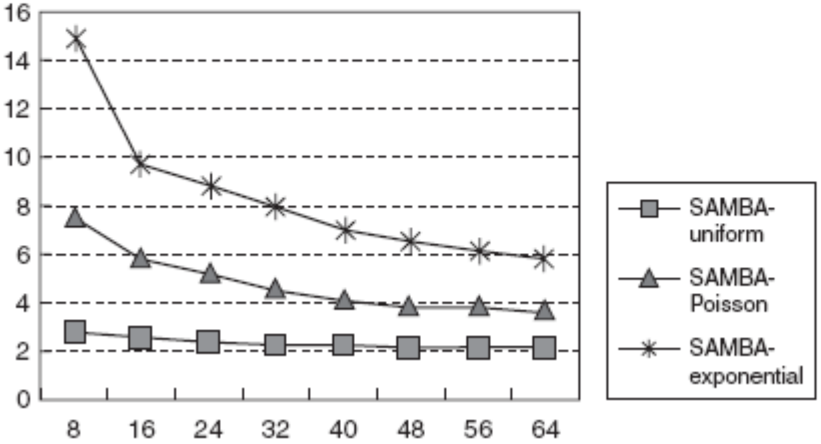


Performance gains of the SAMBA bus

- Note the limitation in the number of modules connected in the busses



Effective bandwidth for buses with different number of modules



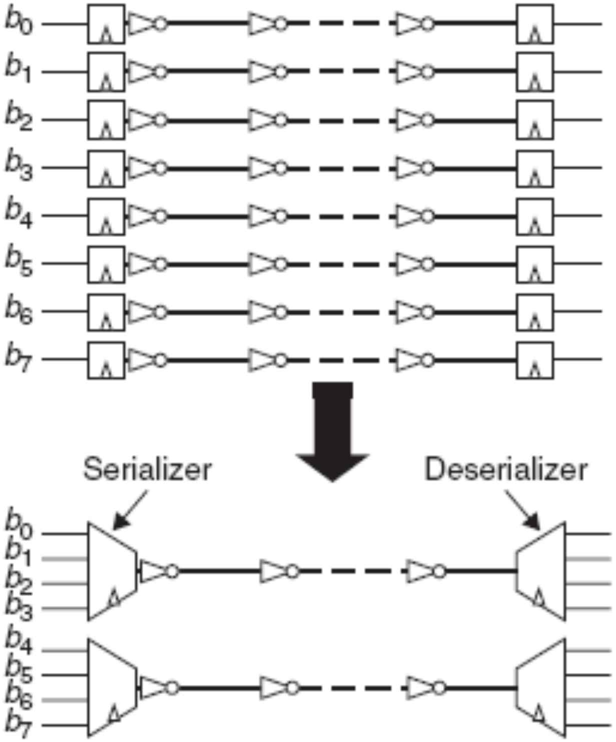
Average latency reduction for buses with different number of modules

SERIAL BUS ARCHITECTURES

- In DSM era, coupling capacitance between adjacent signal lines leads to significant signal propagation delay and power consumption
- The reduction in the number of bus lines results in:

(i) a larger interconnect pitch, which reduces the coupling capacitance

(ii) a wider interconnect, which reduces the effective resistance



SERIAL BUS ARCHITECTURES

- Throughput versus degree of multiplexing

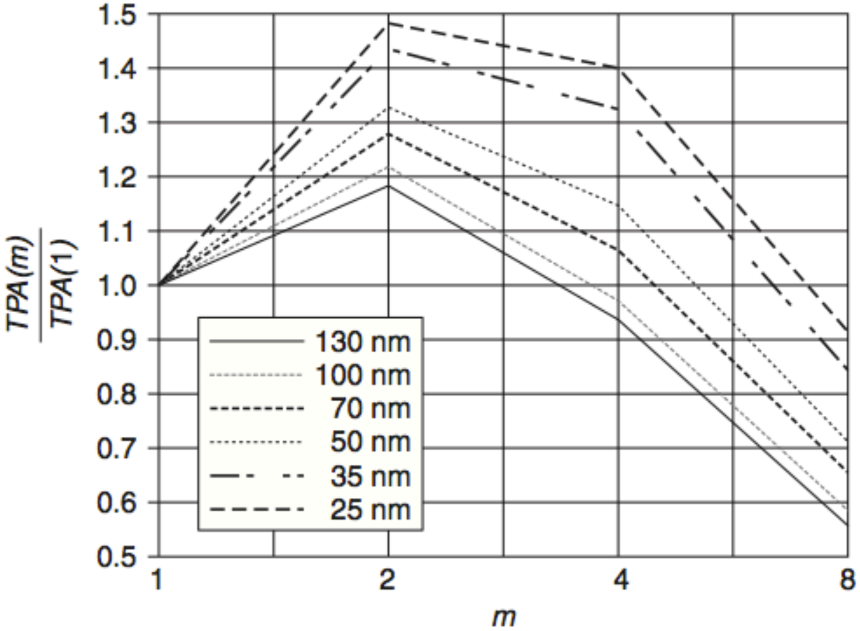
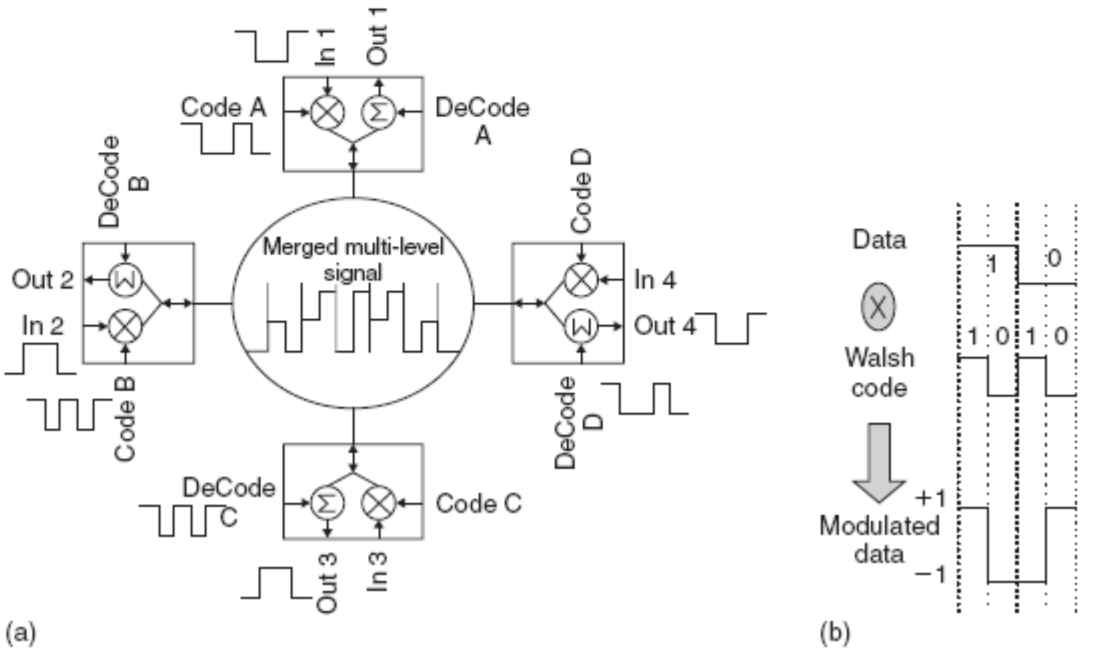


FIGURE 8.13

Relative throughput per unit area of a 64 to (64/m) serial link bus vs. degree of multiplexing *m* [25]
© 2005 IEEE

CDMA-BASED BUS ARCHITECTURES

- Bus: physical interconnect resources are shared in the time domain
- Option: TDMA – again time domain
- CDMA (Code division multiple access): codeword orthogonality, which avoids cross-correlation of codewords and allows perfect separation of data bits modulated with different codewords



CT-Bus

- Hierarchical bus, mixing TDMA with CDMA

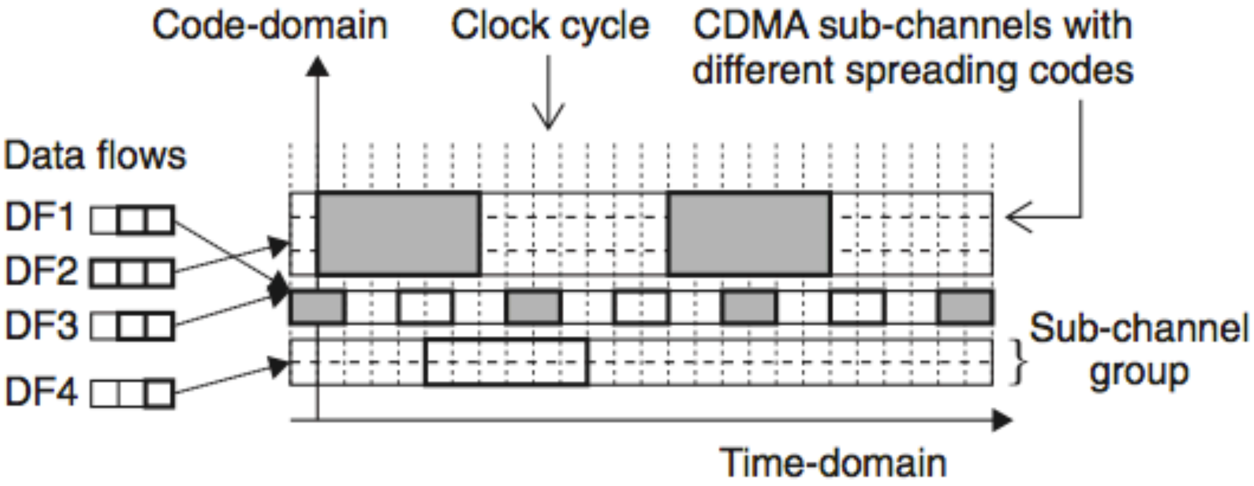


FIGURE 8.16

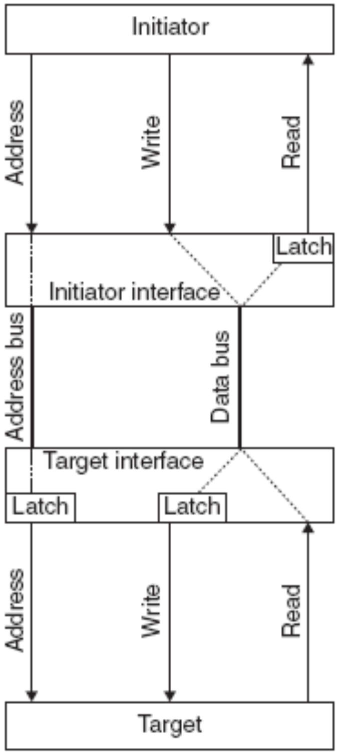
Architecture of CT-Bus (with three CDMA subchannel groups) [30]

© 2004 IEEE

ASYNCHRONOUS BUS ARCHITECTURES

- synchronization occurs using additional **handshake** signals between transfer phases
- lower power consumption compared to traditional synchronous buses
- resilience to clock skew even as the number of IPs (components) connected to the bus increases

MARBLE

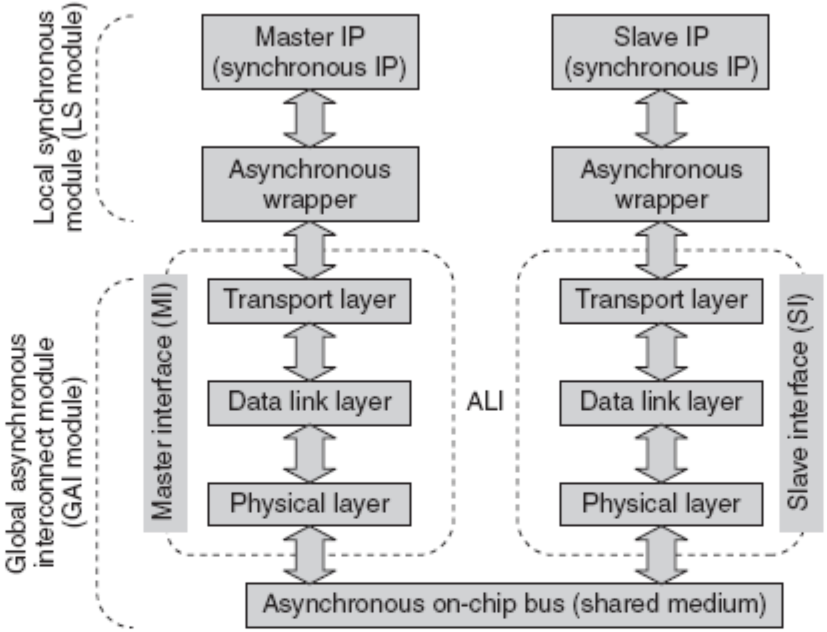


Latches decouple the bus from the components, and free them up for subsequent transfers.

Handshake wires not illustrated

ASYNCHRONOUS BUS ARCHITECTURES

- asynchronous handshake protocol with two-phase signaling and data insensitive (DI) encoding is used for robust and high speed data transfers on the bus
- four-phase signaling and bundled data transfers are used at the IP interfaces for high performance and low complexity.


Local synchronous module (LS module)
Master IP (synchronous IP)
Slave IP (synchronous IP)
Asynchronous wrapper
Asynchronous wrapper
Global asynchronous interconnect module (GAI module)
Master interface (MI)
Transport layer
Transport layer
Data link layer
ALI
Data link layer
Physical layer
Physical layer
Slave interface (SI)
Asynchronous on-chip bus (shared medium)

PERFORMANCE EVALUATION

- SI: single issue – asynchronous
- MI: multiple issue – asynchronous
- MO: multiple issue and OO – asynchronous

} Increased energy consumption

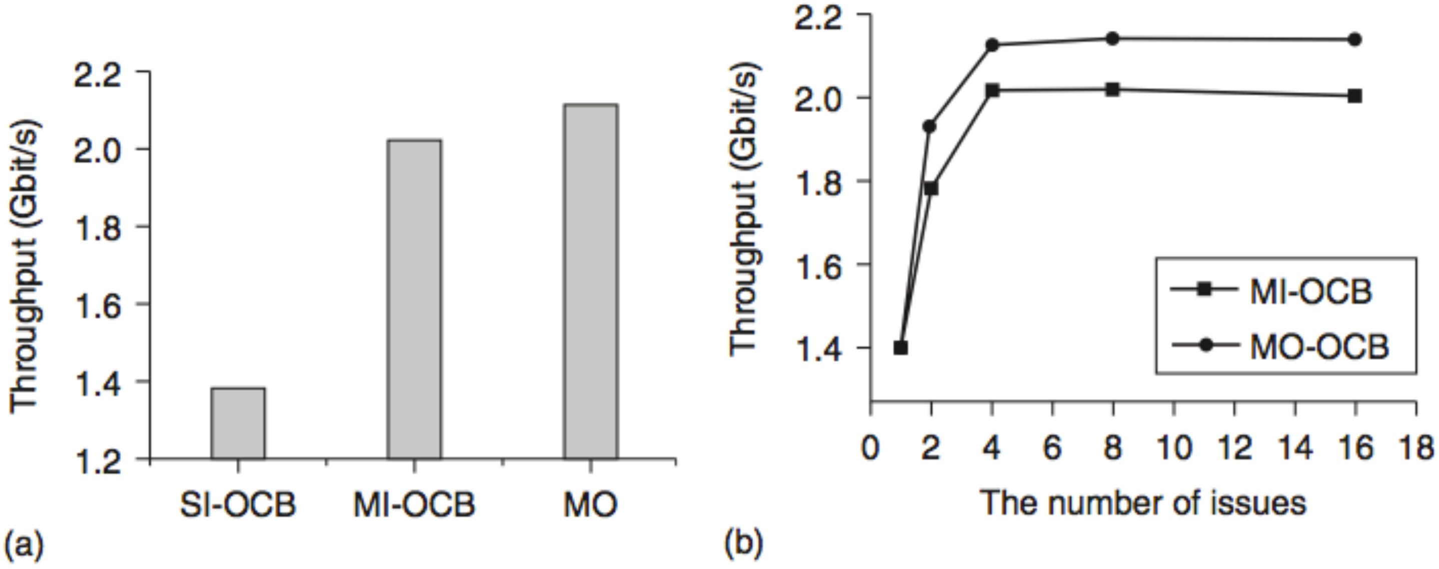


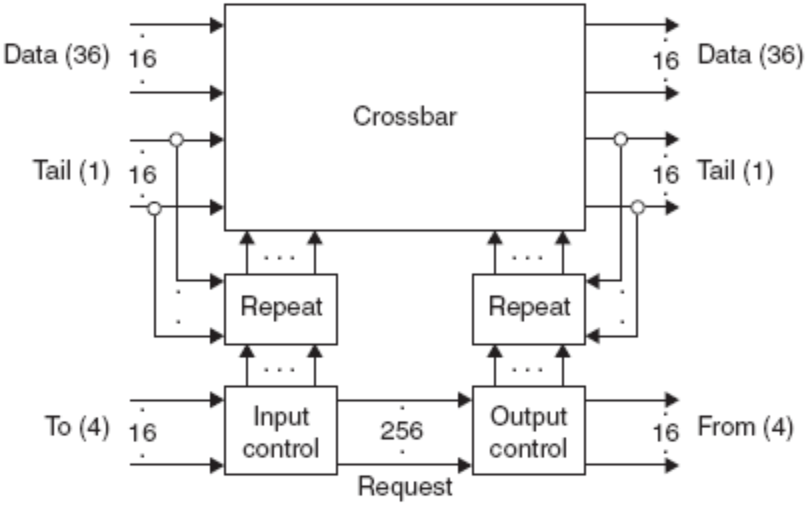
FIGURE 8.20

Simulation results for performance: (a) throughputs (b) throughputs of MI-OCB and MO-OCB as a function of the number of issues [35]

© 2005 ACM Press

NEXUS bus (asynchronous)

- QDI timing model
- *split* channel for each input which specifies which output to send the burst to
- ***merge*** control channel is also required at the output to indicate which input to receive the burst from

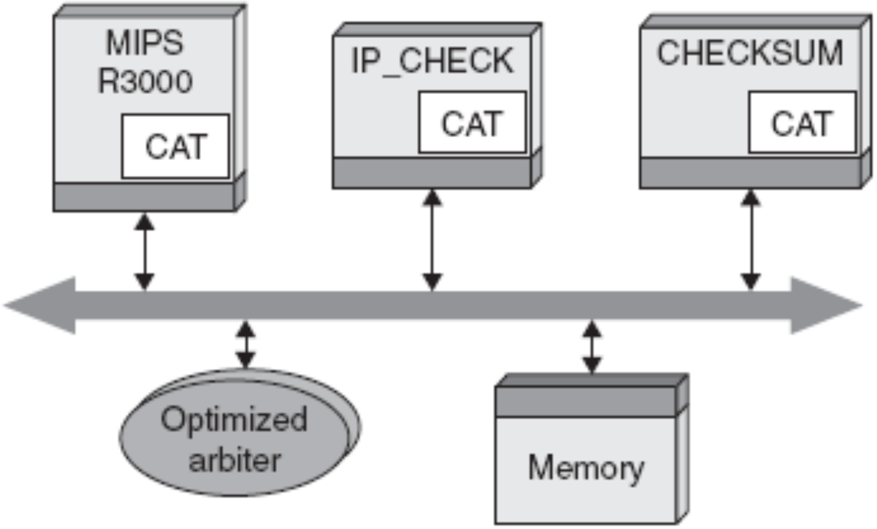


Dynamically Reconfigurable Bus Architectures

- Dynamically reconfigurable bus architectures have the ability to modify certain parameters and even the bus architecture topology dynamically during system execution
- AMBA, Coreconnect – programmable arbitration, TDMA, programmable burst modes

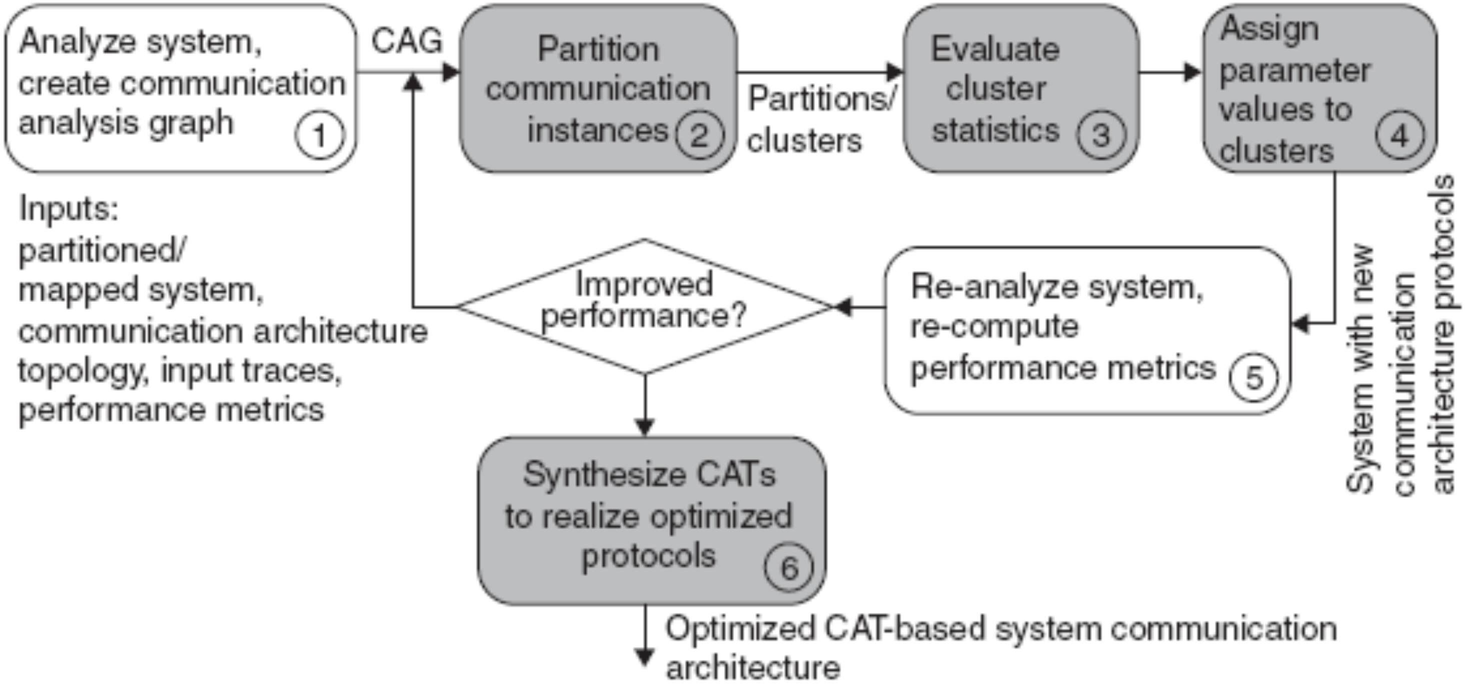
communication architecture tuners (CAT): fixes the arbitration priority according to the packet size

- goal: meet deadlines



CAT – Communication Architecture Tuner

- Methodology to generate de hardware


Analyze system, create communication analysis graph 1
CAG
Partition communication instances 2
Partitions/ clusters
Evaluate cluster statistics 3
Assign parameter values to clusters 4
System with new communication architecture protocols
Re-analyze system, re-compute performance metrics 5
Improved performance?
Inputs: partitioned/ mapped system, communication architecture topology, input traces, performance metrics
Synthesize CATs to realize optimized protocols 6
Optimized CAT-based system communication architecture

CAG: communication analysis graph (CAG)

CAT performance

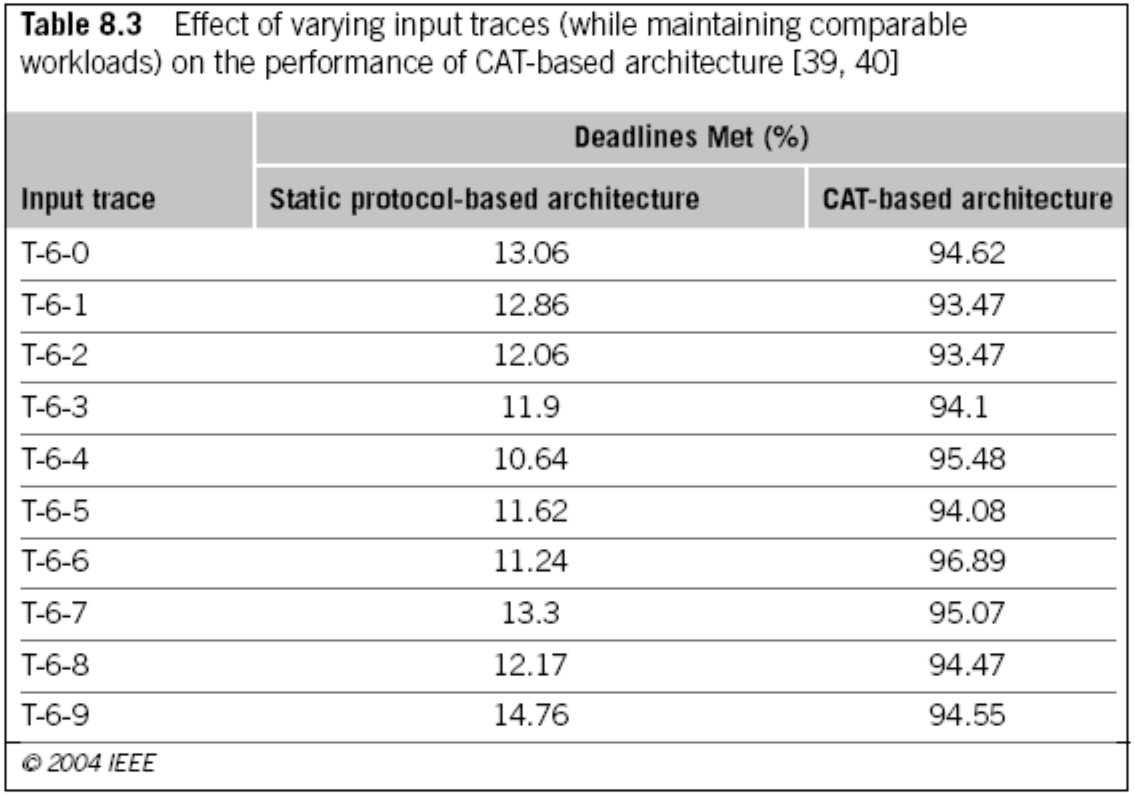
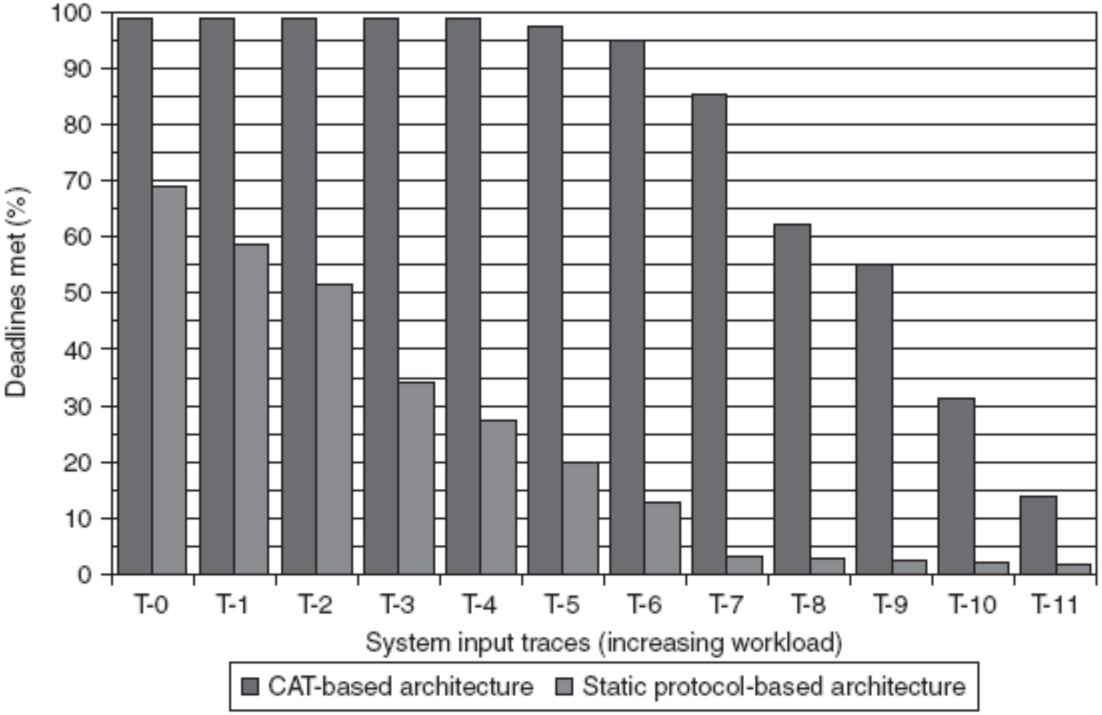


Table 8.3 Effect of varying input traces (while maintaining comparable workloads) on the performance of CAT-based architecture [39, 40]

	Deadlines Met (%)	
Input trace	**Static protocol-based architecture**	**CAT-based architecture**
T-6-0	13.06	94.62
T-6-1	12.86	93.47
T-6-2	12.06	93.47
T-6-3	11.9	94.1
T-6-4	10.64	95.48
T-6-5	11.62	94.08
T-6-6	11.24	96.89
T-6-7	13.3	95.07
T-6-8	12.17	94.47
T-6-9	14.76	94.55

© 2004 IEEE

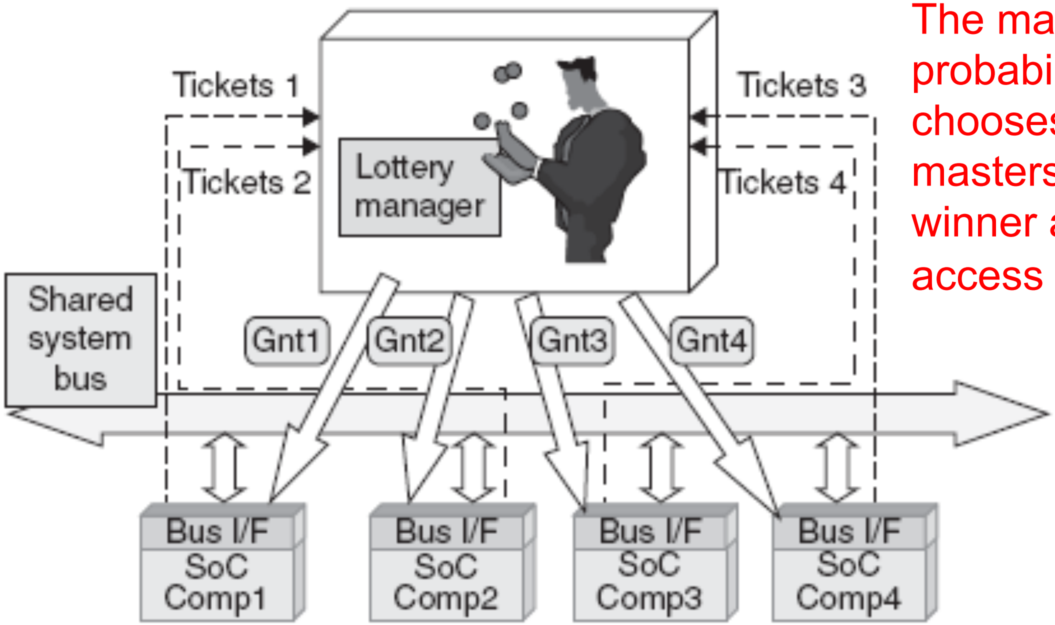
CAT performance



for very low or very high workloads, the gains for the CAT-based architecture are comparatively smaller.

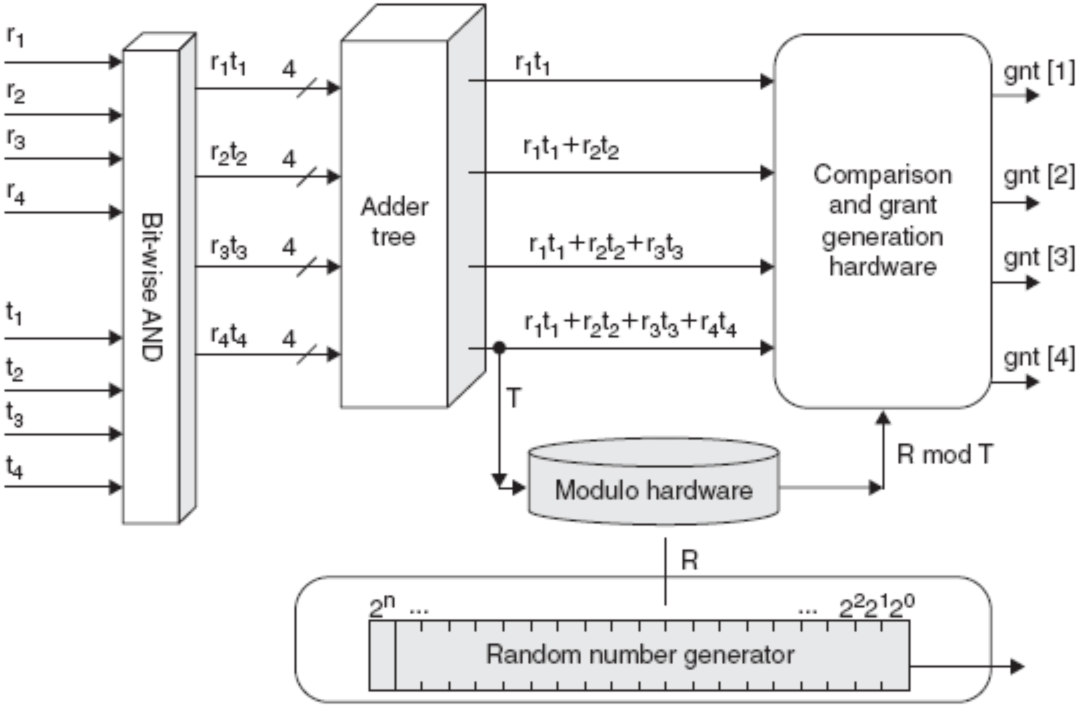
LOTTERYBUS

- Again arbitration
  - Fixed priority: may lead to starvation
  - TDMA: may lead to higher latency values
- LOTTERYBUS: attempts to provide effective bandwidth guarantees, while ensuring low latencies for bursty traffic with real-time latency constraints



The manager probabilistically chooses one of the masters as the winner and grants it access to the bus

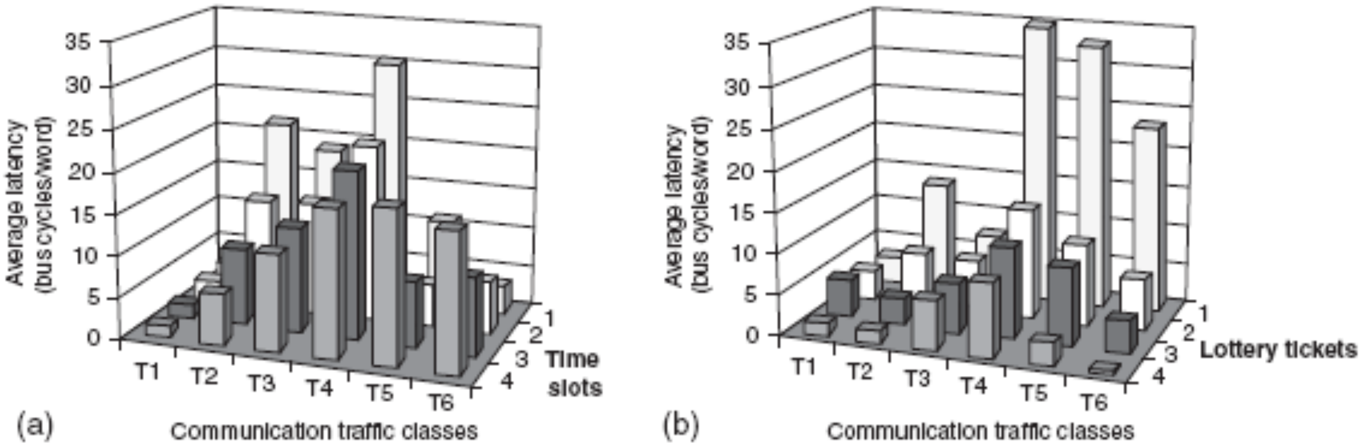
LOTTERYBUS architecture



Implemented in the AMBA bus, with an area increase of 16%

LOTTERYBUS versus TDMA

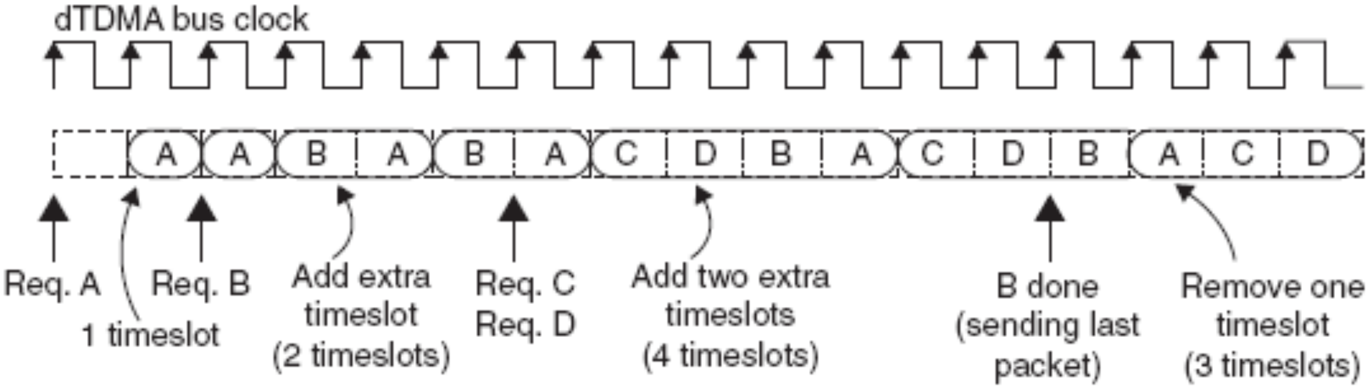
- The communication latency for high priority masters varies significantly for the TDMA architecture (1.65 to 20.5 cycles per word), because the latency of communication in TDMA is highly sensitive to the timing wheel position (i.e., which master's slot currently has access to the bus) when the request arrives.
- The LOTTERYBUS architecture does not exhibit this phenomenon and ensures low latencies for high priority masters.



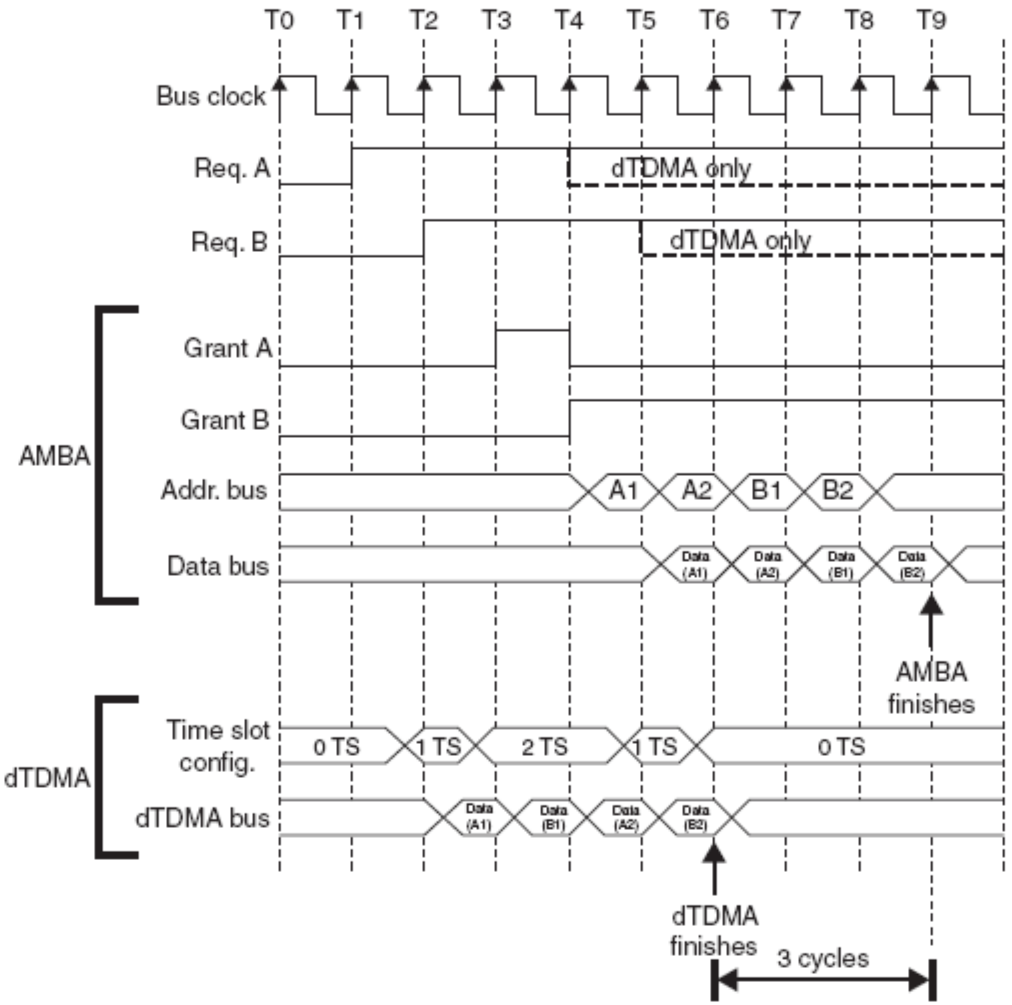
TDMA

dTDMA

- Acts over the burst size
- In dTDMA, the bus arbiter dynamically grows or shrinks the number of timeslots to match the num ber of active transmitters


dTDMA bus clock
A
A
B
A
B
A
C
D
B
A
C
D
B
A
C
D
Req. A
1 timeslot
Req. B
Add extra timeslot (2 timeslots)
Req. C
Req. D
Add two extra timeslots (4 timeslots)
B done (sending last packet)
Remove one timeslot (3 timeslots)

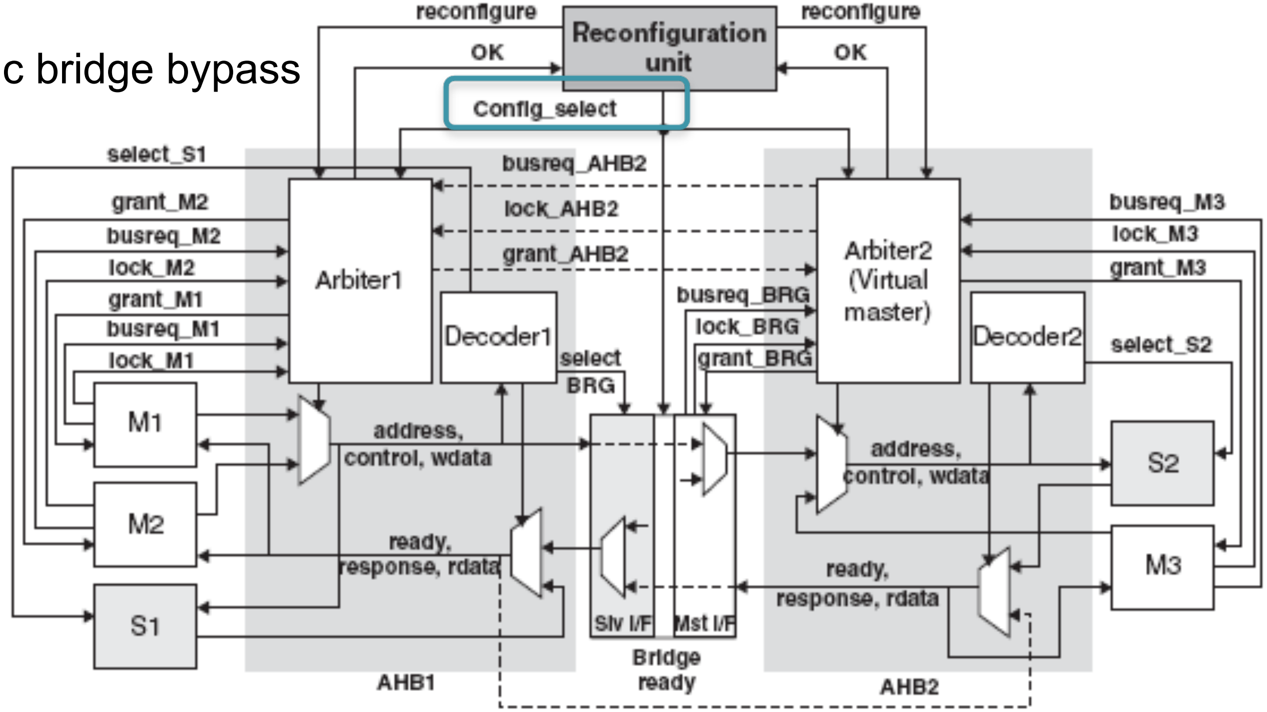
AMBA versus dTDMA



Topology Reconfiguration

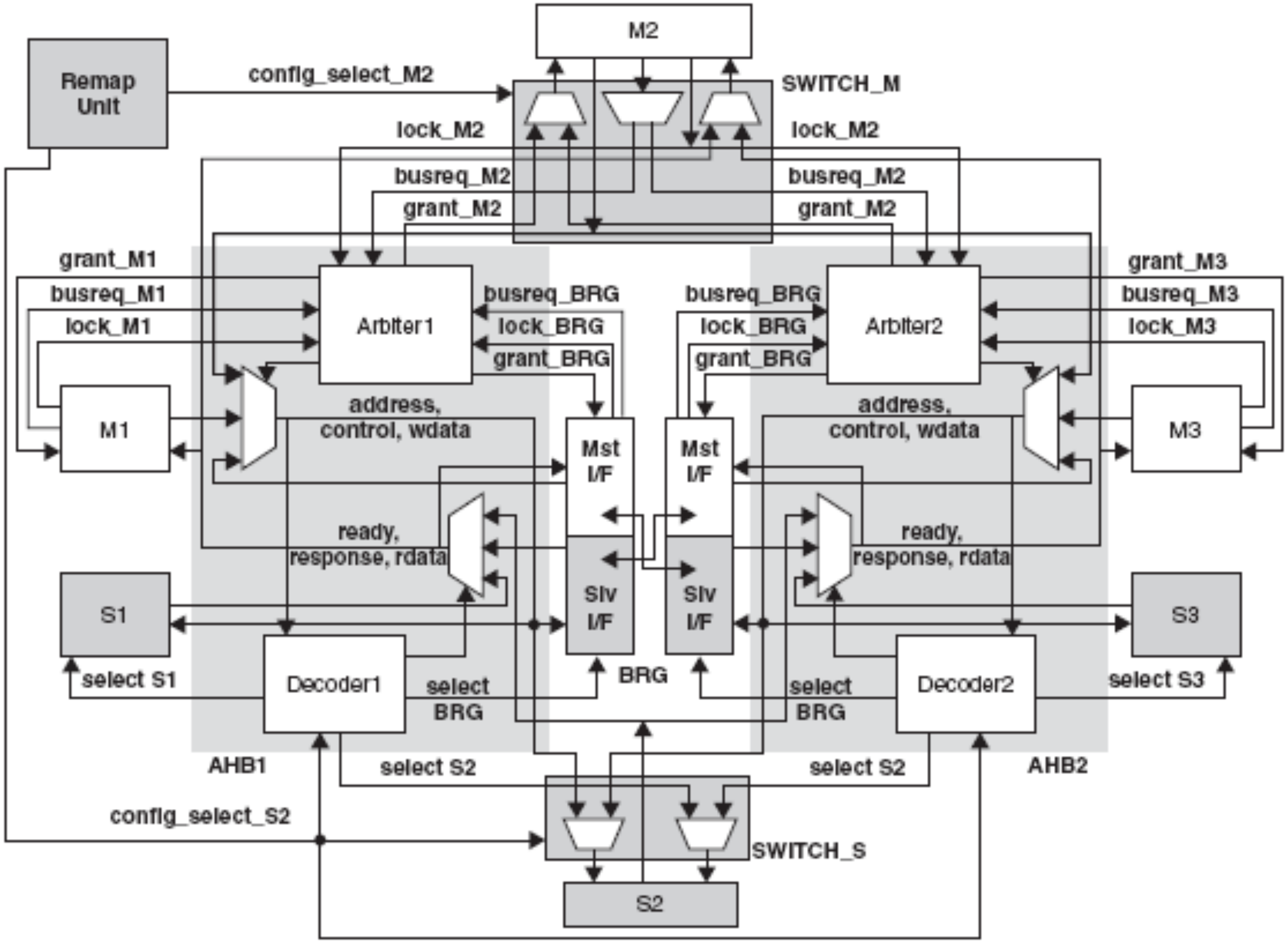
- Example: FLEXBUS
  - Dynamic bridge bypass
  - Dynamic component re-mapping

Dynamic bridge bypass



Dynamic component re-mapping

master M2 and slave S2 can be dynamically mapped to either AHB1 or AHB2.



Dynamic component re-mapping performance

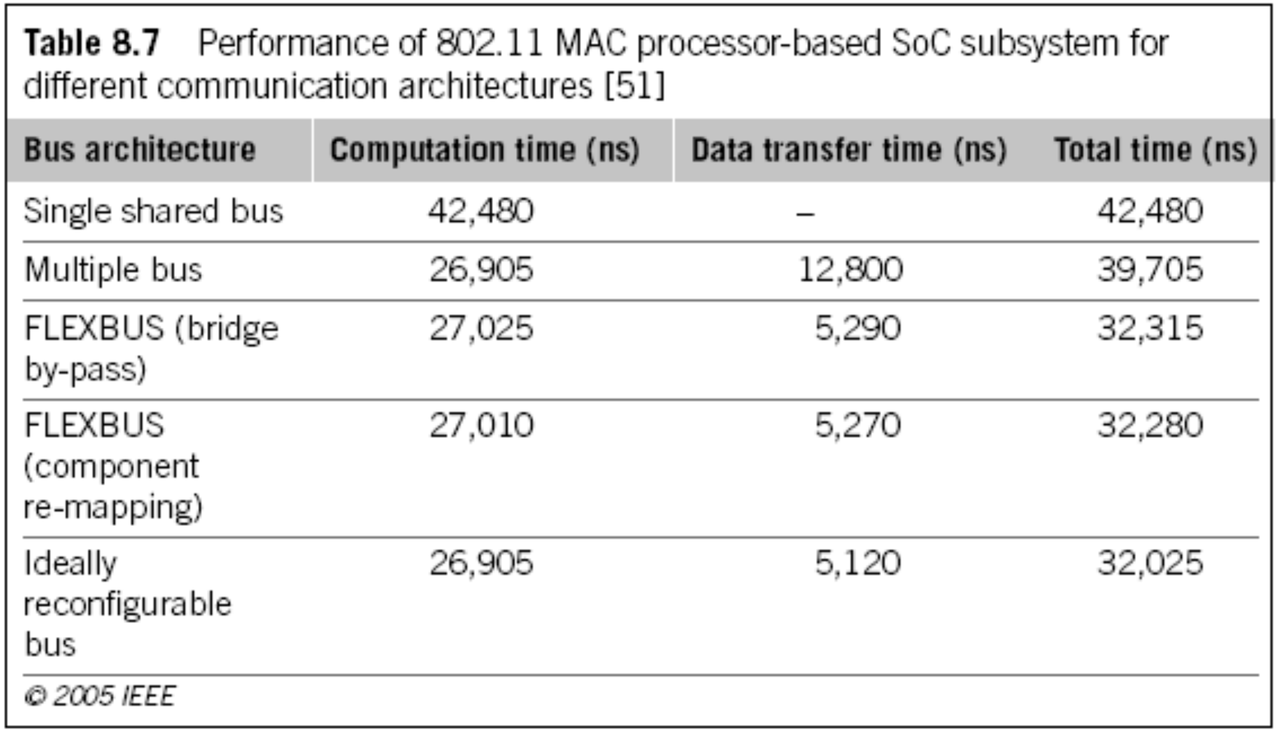


Table 8.7 Performance of 802.11 MAC processor-based SoC subsystem for different communication architectures [51]

Bus architecture	Computation time (ns)	Data transfer time (ns)	Total time (ns)
Single shared bus	42,480	–	42,480
Multiple bus	26,905	12,800	39,705
FLEXBUS (bridge by-pass)	27,025	5,290	32,315
FLEXBUS (component re-mapping)	27,010	5,270	32,280
Ideally reconfigurable bus	26,905	5,120	32,025

© 2005 IEEE