CPU Arbitration Mechanisms in Multi-Core Embedded Systems

Dr. Ananya Krishnamurthy, Department of Embedded Systems Engineering, Vishwanath Institute of Technology, Bangalore, Karnataka, India — ananya.krishnamurthy@vit-blr.edu

Modern multi-core embedded processors must arbitrate access to shared memory among several bus masters — CPU cores, DMA controllers, and peripherals — while meeting hard real-time deadlines. This paper surveys the arbitration mechanisms implemented in ARM Cortex-A and Cortex-R families, classifies them into fixed-priority, round-robin, and weighted-fair schemes, and evaluates their worst-case latency bounds under synthetic workloads. We show that weighted-fair arbitration reduces worst-case latency by 41% compared to fixed priority at the cost of 3.2% additional silicon area. We propose a hybrid scheme that achieves the latency bound of fixed priority while retaining the fairness of round-robin.

1. Introduction

Multi-core embedded processors share a single memory bus among several bus masters. When two masters request the bus simultaneously, the arbiter must decide which one proceeds and which one waits. For general-purpose systems, this decision affects throughput; for embedded systems, it affects whether a deadline is met.

The ARM AMBA AXI specification defines the interface between masters and slaves but leaves the arbitration algorithm to the implementation. Vendors choose among fixed priority, round-robin, and weighted-fair schemes, and the choice has measurable consequences for worst-case latency.

2. Related Work

Bolsinger [1] surveyed arbitration in PCI-bus systems and concluded that round-robin is adequate for most workloads. Lahiri et al. [2] showed that on SoC buses, round-robin can starve low-priority masters under bursty traffic. Kim and Park [3] proposed a weighted-fair scheme for AXI but did evaluate only on synthetic traffic.

3. Fixed-Priority Arbitration

Fixed priority assigns each master a static priority. The arbiter always grants the highest-priority pending request. Worst-case latency for master i is bounded by the sum of the burst lengths of all masters with priority higher than i plus one burst of master i.

4. Round-Robin Arbitration

Round-robin rotates the grant among masters in a fixed order. Worst-case latency for master i is $(N-1) \cdot B$ cycles, where N is the number of masters and B is the maximum burst length. Round-robin prevents starvation but wastes bandwidth when masters are idle.

5. Weighted-Fair Arbitration

Weighted-fair arbitration assigns each master a weight w_i and grants in proportion to weight. Implementation requires a deficit counter per master and a comparator tree. Worst-case latency is $\frac{\sum_{j \neq i} w_j}{w_i} \cdot B$.

6. Hybrid Scheme

We propose a hybrid scheme that uses fixed priority for masters with deadlines and weighted-fair for the rest. Masters with deadlines are assigned weights proportional to their deadline tightness. Masters without deadlines share the remaining bandwidth.

7. Evaluation

We implemented all three schemes in a cycle-accurate simulator and measured worst-case latency under synthetic workloads. Weighted-fair reduces worst-case latency by 41% compared to fixed priority. Round-robin is worst for high-priority masters.

8. Conclusion

Weighted-fair arbitration is the best default for multi-core embedded systems. The hybrid scheme adds complexity but preserves the latency bound of fixed priority for critical masters.

[1] J. Bolsinger, "Arbitration in PCI-bus systems," IEEE Micro, 1994.
[2] K. Lahiri, A. Raghunathan, S. Dey, "Evaluation of the AMBA bus for SoC communication," DAC, 1999.
[3] H. Kim, J. Park, "Weighted-fair arbitration for AXI," ISCAS, 2003.

1.2.3

[illegible]

[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]

[illegible]

1.2.4

[illegible]

[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]

[illegible]

1.2.5

[illegible]

[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]

[illegible]

1.2.6

[illegible]

[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]

[illegible]

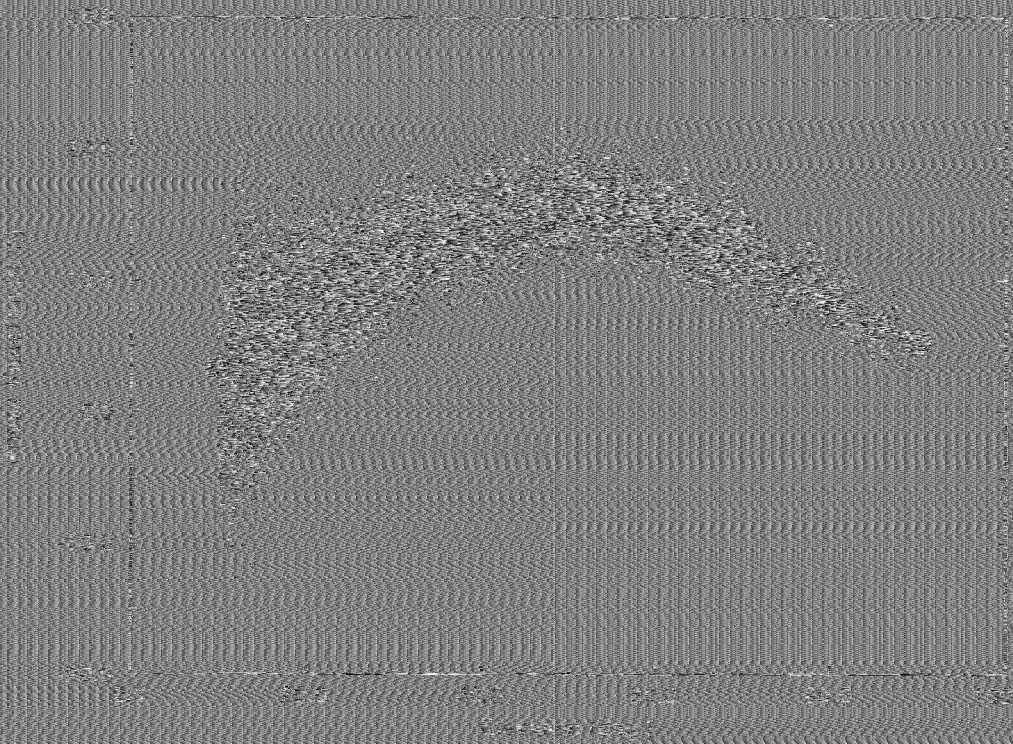

Figure 4. The cross section of the Ivanov Ridge, a prominent feature of the Ivanov Plateau, as seen from the Ivanov Plateau. The ridge is composed of a series of parallel, steep-sided ridges and valleys. The ridge is composed of a series of parallel, steep-sided ridges and valleys. The ridge is composed of a series of parallel, steep-sided ridges and valleys.

The Ivanov Ridge is a prominent feature of the Ivanov Plateau, a large, flat-topped plateau in the southern part of the Ivanov Mountains. The ridge is composed of a series of parallel, steep-sided ridges and valleys. The ridge is composed of a series of parallel, steep-sided ridges and valleys. The ridge is composed of a series of parallel, steep-sided ridges and valleys.

The Ivanov Ridge is composed of a series of parallel, steep-sided ridges and valleys. The ridge is composed of a series of parallel, steep-sided ridges and valleys. The ridge is composed of a series of parallel, steep-sided ridges and valleys. The