Weekly Report

Weiping Liao

1 Performance Model

Our MPSoC is better to be modelled by M/D/1/-/N queue, where N is the number of PEs. When there are k requests in the queue (requests from PEs waiting for main memory access), the arrival rate $\lambda_k = \lambda_0 * (N - k)$, where 0<=k<=N. λ_0 is the memory request rate for a single PE, and equal to $\frac{M}{C_p+M*T_a}$, where M is the number of memory requests, C_p is the cycles consumed by pipeline structure, and T_a is the average memory access time.

Suppose the memory can handle request in the rate of R_m, then the system unitilzation rate $\rho = \frac{R}{R_m}$. The average number of memory requests waiting in the queue W and the average memory access time T_a can be calculated as

$$W = \frac{\sum_{k=0}^{N} k * \rho^k * \frac{N!}{(N-k)!}}{\sum_{k=0}^{N} \rho^k * \frac{N!}{(N-k)!}} \quad (1)$$

$$T_a = \frac{W + 1}{R_m} \quad (2)$$

With the definition $R = \frac{M}{C_p+M*T_a}$, T_a can be decided by the above equation. The performance in terms of IPC is $\frac{I}{C_p+M*T_a}$ where I is the total number of instructions.

This model can be simplified by replacing the factorization with some approximated polynormial representations. But I want to first verify the accuracy of this model before I simplify it. I calculated the IPC by hands for one and two PEs case and it seems the model is very accurate (<2% error). To calculate the performance for more PEs I need to write a program. It should be done easily but I just have not finish that at this point.

2 Multi-core Optimization

The following is the outline I am thinking:

Problem formulation: given fixed area, design a multi-core architecture to maximum performance. This is the first step and further extensions are listed later in this report.

Approaches: (1) We list all architectural components with possible configuration. For example, caches with different size and associativity, different number of ALUs, etc. For each configuration, we decide (A) access latency and (B) area. (2) For each PE core, we put together possible configuration and consider the intra-core interconnect, as Changbo and Luke did. (3) The total number of PEs is decided by the area constraint. (4) We put together all cores and consider the inter-core communication, for example bus. (5) We use TPWL to quickly explore the whole design space. By this approach, we automatically consider the single ISA heterogeneous multi-core design.

Note simply use TPWL for each single core configuration is pointless because the optimal result for one single core may not necessarily be the one in the optimal multi-core design.

The step should be easy to achieve. The follow-up extensions include: (1) consider power consumption and target at given power envelop; (2) consider network-on-chip instead of bus; (3) study the impact of technology scaling on multi-core design; (4) consider dynamic power and thermal management and the cost of on-chip voltage supply.