Modulo Scheduling with Regular Unwinding

Benoît Dupont de Dinechin

March 30, 2004

We introduce a new technique for modulo scheduling, based on the unwinding of the modulo scheduling problem, and the acyclic scheduling of the unwinded problem under an additional constraint of regularity. Given $\lambda$, the modulo schedule initiation interval, a regular unwinded schedule is such that two successive instances of any operation are scheduled at least $\lambda$ cycles apart. For a given $\lambda$, we establish the equivalence between the modulo schedules, and the regular unwinded schedules of suitable size.

A main benefit of the regular unwinding technique is the re-formulation of the modulo scheduling problems in the classic framework of acyclic scheduling. In particular, we introduce new modulo scheduling problem relaxations that are solvable in pseudo-polynomial time. These results are obtained by combining regular unwinding with the time-constrained instruction scheduling relaxation of Leung, Palem & Pnueli [2].

Modulo scheduling [3, 1] is an instruction scheduling technique used for software pipelining inner program loops. In modulo scheduling problems, a set of operations $\{O_i\}_{1 \leq i \leq n}$ is repeatedly executed with a period of $\lambda$ cycles, the initiation interval. Precisely, we denote $\{\sigma_i\}_{1 \leq i \leq n}$ the schedule dates, and the execution is constrained as follows:

- Uniform dependence constraints denoted $O_i \xrightarrow{\alpha^I_{ij}, \beta^I_{ij}} O_j$: for each such dependence, a valid modulo schedule satisfies $\sigma_i + \alpha^I_{ij} - \lambda \beta^I_{ij} \leq \sigma_j$. The latency $\alpha^I_{ij}$ and the distance $\beta^I_{ij}$ of the dependences are non negative integers. The carried dependences are such that $\beta^I_{ij} > 0$.

- Modulo resource constraints: each operation $O_i$ requires $\bar{b}_i \geq 0$ cumulative resources for all the time intervals $[\sigma_i + k\lambda, \sigma_i + k\lambda + p_i - 1], k \in \mathbb{Z}$, and the total resource use at any time
must not exceed $\vec{B}$. The positive integer value $p_i$ is the processing time of operation $O_i$.

References

