Forced-Directed List-Scheduling

Vytaras Juraska, Electronic Engineering 6th Semester

Why Force-Directed List-Scheduling exists

- Big drive forwards was Design Space Exploration
- Combining best of both scheduling approaches
- Unique constraints, which were still improvable

Which approach focuses on which constraints

- List-Scheduling (LS) - minimise execution time having hardware constraints

- Force-Direct (FD) - minimise scheduling problems caused by having hardware constraints while also having a fixed latency constraints

- Force-Directed List-Scheduling (FDLS) - aiming for as close as possible to the most optimal execution time having hardware constraints and limited time frame of execution

Relevant parts of List-Scheduling

Relevant parts of LS

- Uses a ready-list - a list of operations ready to be executed

- Has a prioritization based on mobility of tasks

- Mobility: $\mu_i = t_i^L - t_i^S; i = 0, 1, \ldots, n.$

- High mobility tasks are deferred

Relevant parts of Force-Direct

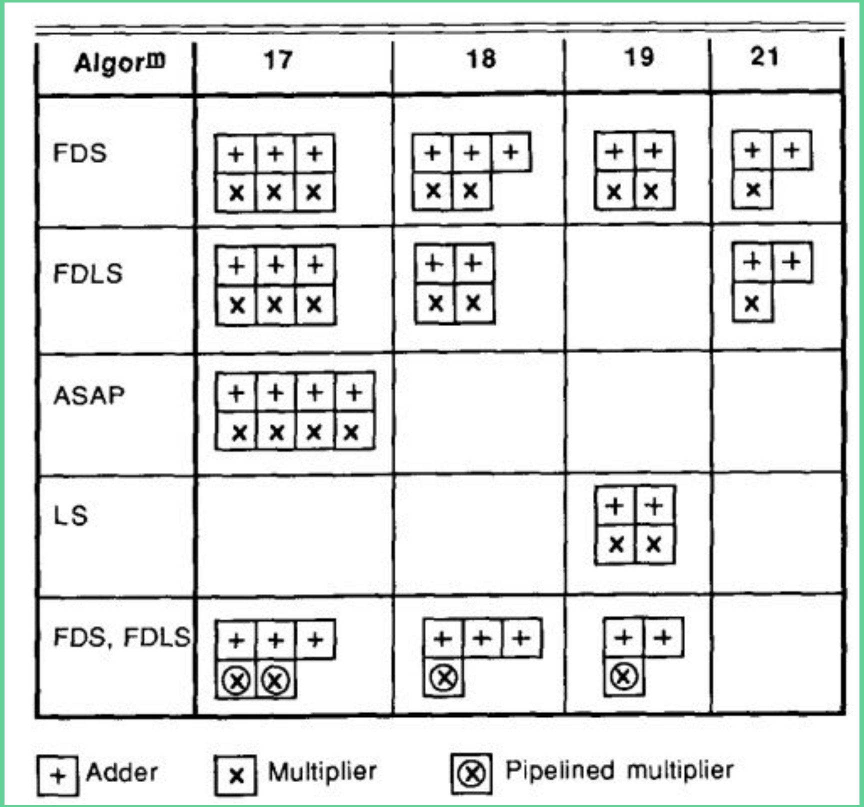
Relevant parts of FD

- Probability prefers little to none mobility of the task
- Concurrency is defined as force, which is calculated individually for each unscheduled task, each time a specific task gets scheduled.
- Two types of forces: self-force and predecessor-successor force
- Using force the goal of lowering concurrency is achieved

Concept of FDLS

- Main idea: adapt FD forces into a process of deferring a task, similar to LS
- Instead of mobility, the priority is determined by a force
- Tasks are not necessarily pushed to the next step, when they are deferred.

Benchmarking the scheduling concepts



Algorm	17	18	19	21
FDS	+ + + × × ×	+ + + × ×	+ + × ×	+ + ×
FDLS	+ + + × × ×	+ + × ×		+ + ×
ASAP	+ + + + × × × ×			
LS			+ + × ×	
FDS, FDLS	+ + + ⊗ ⊗	+ + + ⊗	+ + ⊗	

+ Adder   × Multiplier   ⊗ Pipelined multiplier

Quick summary of FDLS

- Essentially low computational power needed for acceptable performance

- Task concurrency is evaluated globally, before a deferral happens, to meet the resource constraints

- Bigger scheduling problems start introducing difficulties, since force calculation is quite complex in bigger applications

- Since all of the mentioned scheduling approaches are heuristic, they are not going for absolute maximum performance, hence the results can be more time consuming

Resources

1. P. G. Paulin and J. P. Knight, “Force-directed scheduling in automatic data dath synthesis,” in Proceedings of the 24th ACM/IEEE Design Automation Conference , 1987, pp. 195–202.

2. R. A. Walker and S. Chaudhuri, “Introduction to the scheduling problem,” IEEE Design & Test of Computers, vol. 12, no. 2, pp. 60–69, 1995.

3. G. De Micheli, Synthesis and optimization of digital circuits. McGraw Hill, 1994, no. BOOK.

4. P. G. Paulin and J. P. Knight, “Force-directed scheduling for the behavioral synthesis of asics,” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 8, no. 6, pp. 661–679, 1989.

Thank you!