

Partitioning - I

Introduction to Partitioning

Mahapatra-Texas A&M-Fall'00

1

Synthesis

- Why synthesis?
 - Synthesis converts behavioral process into customized digital-hardware processor.
- Problems with large behavioral processes:
 - synthesis runtime lasts tens of hours or more
 - resulting processor may have high power consumption
 - package size may exceed the constraints

Mahapatra-Texas A&M-Fall'00

2

Possible remedies

- ♠ synthesis runtime lasts tens of hours or more
 - use heuristic and trade of the quality
- ♠ resulting processor may have high power consumption
 - isolation of processor sub-circuit to avoid unnecessary signal switching
- ♠ package size may exceed the constraints
 - structural partitioning of processor

Mahapatra-Texas A&M-Fall'00

3

System partitioning

System level partitioning problem



Assignment of operations to
hardware or software

- Assignment of an operation to HW or SW determines the *delay of the operation*
- Assignment of operation to a processor and to more application-specific HW circuits involve additional *delays due to communication* overhead.

Good partitioning scheme \Rightarrow *Minimize this communication*

Mahapatra-Texas A&M-Fall'00

4

System partitioning contd..

- Increasing operations in software on a single processor
⇒ increases processor utilization
- **system performance:** depends on hw-sw partition on utilization of processor and bandwidth of bus between processor and application specific hardware.
- **Characteristic of Partitioning scheme:** capture and make use of partition's effect on system performance in making trade-off between hw and sw implementation of an operation.
 - Devise a “*partition cost function*”.

Mahapatra-Texas A&M-Fall'00

5

Partitioning

- **Cost function:**
 - Directs the partitioning algorithm towards desired solution
 - optimum solution is minimum cost function
- **Need to capture:**
 - effects of size of hw/sw parts
 - effects on timing behavior of these portions on cost function (contrast: optimized area/pinout)
 - difficult due to the problem being global in nature
 - approximation is used to account the effect on total latency

Mahapatra-Texas A&M-Fall'00

6

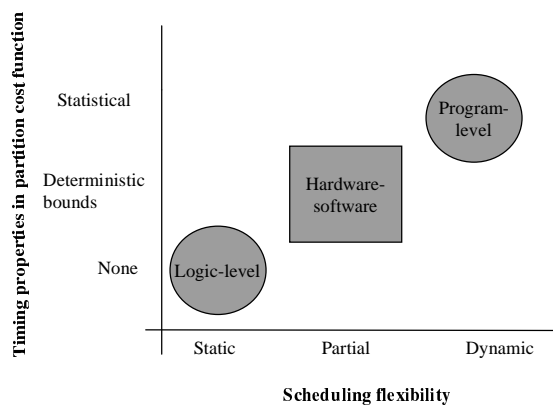
Partitioning

- Partitioning in **software**: extensive use of statistical timing properties to drive partitioning algorithm.
 - *Dynamic or runtime, excess time., flexible*
- Partitioning in **hardware**: attempts to divide circuits that implement schedule operations.
 - *Static, less time, non-flexible*
- An intermediate approach is advised: incrementally computable of cost function f .
 - *partial, deterministic bound on timing properties,*

Mahapatra-Texas A&M-Fall'00

7

Timing properties in partition cost function

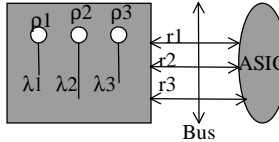


Mahapatra-Texas A&M-Fall'00

8

A Partitioning cost function

- Consider software model in terms of set of program threads and cost function f .



- where, λ_i (per second) is thread latency: execution delay
 - ρ_i (per second) thread reaction rate: invocation rate of the program thread
- processor utilization P is calculated by $P = \sum_{i=1}^n \lambda_i \rho_i$
- Bus utilization B (per second) = $\sum_{j=1}^m r_j$..m variables to be transferred, r_j = inverse of minimum time interval between consecutive samples for variable r_j .

Partition cost function

- Software characterization using λ , ρ , P and B : *static bound*
 - can be used to select appropriate partition of system functionality between hardware and software.
- over estimation of processor and bus bandwidth is possible (since actual distribution of data communication is not captured above)
- Include S_H (*hardware size*) bottom up.
- Characterize interface using set of communication ports (one per variable)
 - overhead due to communication between hw and sw is manifested by the utilization of bus bandwidth.

Partitioning with cost function

- From a given set of sequencing graph models and timing constraints, create two sets of sequencing graph models such that one can be implemented in hw and the other in sw and the following is true:
 - timing constraints are satisfied for the two sets of graph models
 - processor utilization, $P \leq 1$
 - bus utilization, $B \leq B'$
 - A partition cost function, $f = f(S_H, B, P^{-1}, m)$ is minimized.

Partitioning using heuristics

- Minimum cost function can be achieved by trying very large number of solutions (exponential relation to number of operations)
 - *heuristics are used for good solution that may show minimum cost function for some local properties*
- Start with constructive *initial solution* on which iterative procedure can be applied to improve the solution
 - exchange operations or paths between partitions, apply procedure
- A good heuristic is relatively insensitive to initial solution
 - exchange of large number of operation makes it more insensitive to starting solution

More to continue

- More on functional partitioning
- Algorithms used for partitioning
- Assignments:
 - partitioning some benchmark problems in Ptolemy environment.
- Current reference:
 - Rajesh Gupta et.al. “Hardware-Software Cosynthesis for Digital Systems”, IEEE Design & Test of computers, Sept 1993.