APPLICATION OF HIGH-LEVEL MEMORY SIZE ESTIMATION FOR GUIDANCE OF LOOP TRANSFORMATIONS IN MULTIMEDIA DESIGN

Per Gunnar Kjeldsberg
Norwegian University of Science and Technology
Trondheim, NORWAY

Francky Catthoor
IMEC, Leuven, BELGIUM
Also at EE. Dept. of Kath. Univ. Leuven

Einar J. Aas
Norwegian University of Science and Technology
Trondheim, NORWAY

We demonstrate how a novel technique for high-level memory requirement estimation can be used in system level synthesis for data-dominated multimedia applications. Using a polyhedral description of partitioned arrays and their dependencies, guiding hints for the loop ordering are presented to the designer. Together with estimates on the upper and lower bounds of the memory size requirement with a partially fixed execution ordering they are used in the early system design trajectory to find an implementation with low memory requirement. The methodology is demonstrated using a representative multimedia application.

Motivation and Context

For data dominated HW/SW systems:

- Data transfer and storage determine cost and performance parameters.
- Must be main focus of the designer to achieve cost-optimized end product [Catthoor98].

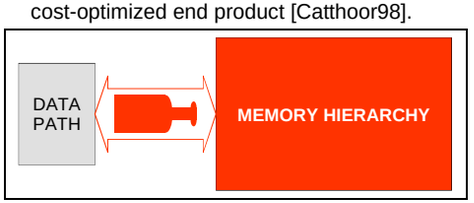

Figure 1: Data dominated embedded system

- At system level no detailed storage requirement information is available.
- Guiding and estimation techniques are essential.

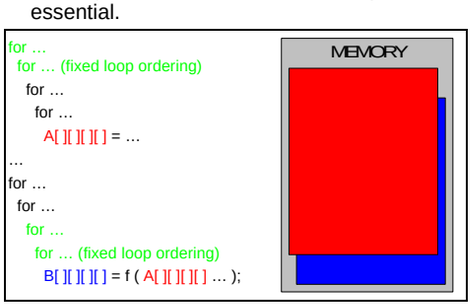

Figure 2: High-level code with complex multi-dimensional loop nests and arrays

- Accurate estimates must take in-place mapping possibilities into account.

Definitions

Dependency Part (DP), see Figure 3.

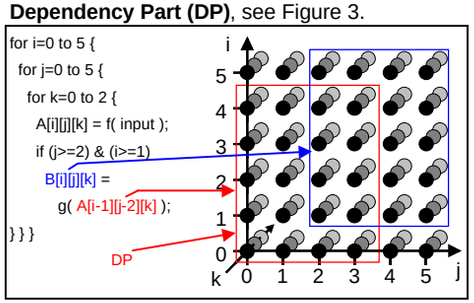

Figure 3: Dependency Part

Dependency Vector (DV), see Figure 4.

Dependency Vector Polytope (DVP), see Figure 4.

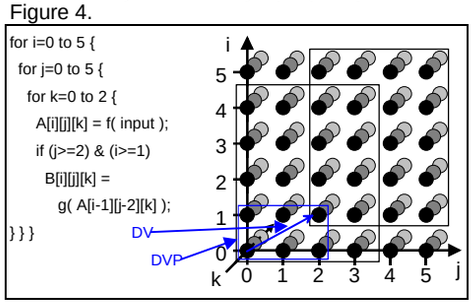

Figure 4: Dependency Vector and Dependency Vector Polytope

Spanning/Nonspanning Dimensions (SD/ND), in Figure 4: SD={*i*,*j*}, ND={*k*}.

Estimation Methodology

Generate data-flow graph reflecting data dependencies in the application code

↓

Estimate upper and lower bounds on dependency sizes per signal based on partly fixed execution ordering

↓

Traverse the data-flow graph to find the total size of simultaneously alive dependencies

=

Bounds on the application's storage requirement

Estimate with no execution ordering fixed:
Upper Bound (UB) = Size (DP) - Overlap = 36
Lower Bound (LB) = Size (DVP) - Overlap = 5

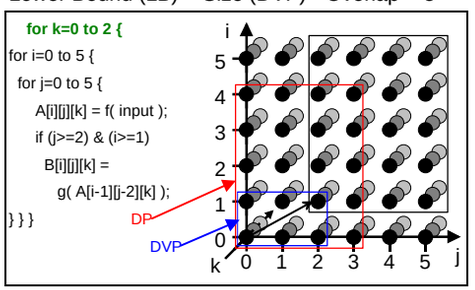

Figure 5: Nonspanning Dimension k fixed outermost

Estimate with ND *k* outermost:
DP reduced → UB=12. DVP unchanged → LB=5. See Figure 5.

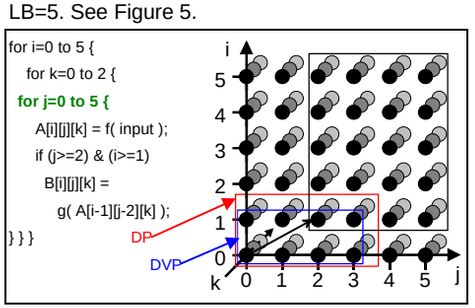

Figure 6: Spanning Dimension j fixed innermost

Estimate with SD *j* innermost:
DP reduced → UB=18. DVP extended → LB=6. See Figure 6.

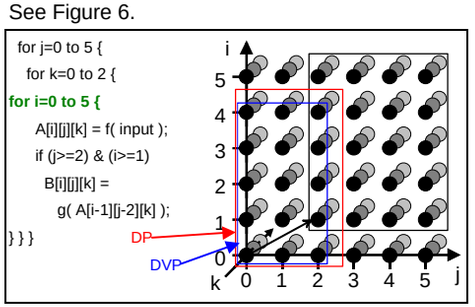

Figure 7: Spanning Dimension i fixed innermost

Estimate with SD *i* innermost:
DP reduced → UB=33. DVP extended → LB=11. See Figure 7.

Guiding Hints for Loop Ordering

Fix:

- Nonspanning Dimensions outermost
- Spanning Dimensions innermost
- Among Spanning Dimensions fix:
  - o Dimensions with shortest distance between border of DVP and border of DP innermost

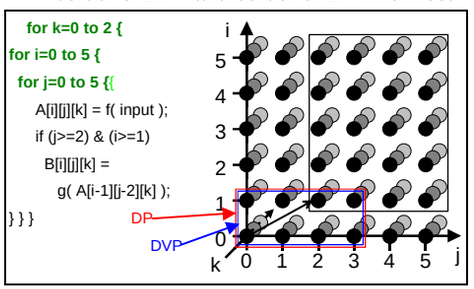

Figure 8: Ordering according to guiding hints; k outermost, j innermost and I second innermost

Estimate with ordering according to hints:
DP reduced → UB=6. DVP extended → LB=6. See Figure 8.

MPEG-4 Motion Estimation Kernel

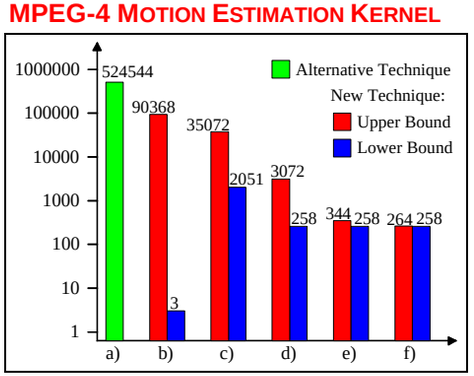

Figure 9: Total storage requirement a)Multiplication of array dimensions b) No ordering, c) y_p outermost, d) y_s outermost, e) x_s second outermost, f) y_p third outermost and x_p fourth outermost

Presented at the IEEE Nordic Signal Processing Symposium, Kolmården, Sweden, June 2000

imec NTNU