The Heterogeneous Structure Problem in Hardware/Software Codesign: A Macroscopic Approach.

J.A. Maestro, D. Mozos, R. Hermida
Dpto. de Arquitectura de Computadores y Automática
Universidad Complutense - 28040 Madrid, Spain
E-mail: maestro@eucmax.sim.ucm.es

Abstract

As the Codesign problems become more and more complex, characterizing the scheduling and allocation details of the tasks with macroscopic magnitudes easy to handle, can help to solve them in an efficient way.

1.- Introduction.

In recent years, and due to the embedded system market expansion, ***Codesign*** [1] has become a major research trend, as the natural continuation of High Level Synthesis. In [2], we proposed a change in the way of tackling the estimation process, raising the level of abstraction and using new methodologies from a different point of view. That is what we call a ***macroscopic approach***, and it is characterized by the introduction of *more abstract data, that can substitute the complex tables of operators and functional units in High Level Synthesis.*

Many previous systems in literature [3,4] lack certain advantages that the macroscopic approach has. In this paper, *we have expanded the model with the inclusion of new macroscopic variables*, and the redefinition of the classical concept of overlapping.

2.- The density factor.

In classical approaches, *the probability of sharing functional units between two nodes was inversely proportional to their overlap*. However, *that is not always true*, as the actual distribution of the functional units can produce a stronger conflict when having a quasi-sequential execution than in the case of parallel execution.

Therefore the inner distribution of units is relevant, and must be considered when calculating the sharable hardware of two nodes. In this way, we have introduced a macroscopic parameter to characterize this fact, called the *density factor*, δ. This factor, $\delta_{i,S(i)}$, expresses, as a real function, the cost distribution of a node *i* along its execution range, assumed a certain implementation, *S(i)*. The mentioned function is a real polynomial, that indicates how much cost is necessary to implement the functionalities in every point. Although it may seem strange to use a continuous function, this choice has several advantages: the data are simpler to handle and the processes are much easier. Besides, evolving to a continuous function when increasing the abstraction level is a natural step, as details begin to fade out at this point. The decision of how many points of the cost distribution should be considered relies on the designer.

3.- Redefinition of overlapping.

Once the density factor has been defined, it is necessary to redefine the concept of overlapping factor, σ [2]. Now, two nodes will overlap more, not when their execution ranges take place at the same time, but when their regions with a high operational load are executed simultaneously.

The classical definition only considered the physical intersection of the two nodes, regardless of their inner structure:

$$\sigma_{i,j} = \frac{t_{end}(j) - t_{st}(i)}{t_{end}(i) - t_{st}(i)}$$

However, as it was stated in the previous section, the inner distribution of operations is critical in determining the overlapping factor. In other words, it is not only important how much two nodes overlap, but also with which *intensity*, defined in this way (see Figure 1):

$$\iota_{i,j}^{\alpha} = \int_{o}^{c} \delta_{i,S(i)}(t) \cdot \delta_{j,S(j)}(t+\alpha) \cdot dt \qquad (1)$$

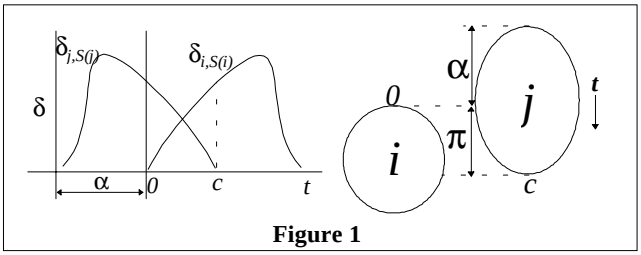

Figure 1

The redefinition of the overlapping factor σ, which should be comprised in the range [0,1], is achieved by calculating a maximum *threshold* of the intensity, *T:*

This work has been granted by CICYT TIC 96/1071 and HCM CHRX-CT94-0459

$$\sigma_{i,j} = \frac{\iota_{i,j}}{T} \;\; ; \;\; T = \int_{t_{st}(i)}^{t_{end}(i)} \underset{dom(j)}{MAX}[\delta_{j,S(j)}(t)] \cdot \delta_{i,S(i)}(t) \cdot dt \qquad (2)$$

4.- Experimental results.

The first experiment in this section will be to study the effect of the overlapping factor on a system composed of two nodes, *i* and *j* (Figure 2).

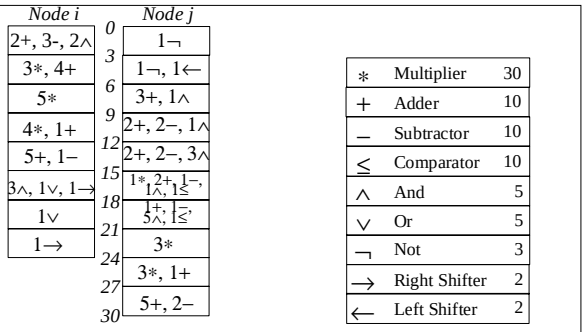

Figure 2

∗	Multiplier	30
+	Adder	10
−	Subtractor	10
≤	Comparator	10
∧	And	5
∨	Or	5
¬	Not	3
→	Right Shifter	2
←	Left Shifter	2

Figure 3

Node *i* is assumed to have an execution time of 24, and *j* of 30. The cost of every functional unit is shown in Figure 3. Studying the parallelism of the different kinds of these units, the associated costs of both nodes has been found to be 252 and 200. Then, a similarity of $\kappa_{[j,S(j)],[i,S(i)]}$ = 0.8750 is obtained. Let us assume in this example that π=15, which implies that α=15. (See Figure 1).

a)*Traditional approach*. As it was defined in section 3, the classical definition of the overlapping factor is:

$$\sigma_{i,j} = \frac{t_{end}(j) - t_{st}(i)}{t_{end}(i) - t_{st}(i)} = \frac{15}{24} = 0.6250$$

b) *Macroscopic approach*. Now we consider the inner structure of both nodes to find out the intensity of their overlapping. The set of extremes that will represent the nodes are {(6,150), (21,2)} for node *i*, and {(0,3), (24,100)} for node *j*. Therefore, a 3-degree polynomial has been chosen for both cases:

$\delta_{i,S(i)} = 0.0877 \cdot t^3 - 3.5520 \cdot t^2 + 33.152 \cdot t + 60.0160$

$\delta_{j,S(j)} = -0.0140 \cdot t^3 + 0.5052 \cdot t^2 + 3$

Next, the overlapping factor is calculated by (2):

$$\iota_{i,j}^{15} = \int_{o}^{15} \delta_{i,S(i)}(t) \cdot \delta_{j,S(j)}(t+15) \cdot dt = 160259.6$$

$$T = \int_{0}^{24} 100 \cdot \delta_{i,S(i)}(t) \cdot dt = 189506.5$$

$$\sigma_{i,j} = \frac{\iota_{i,j}}{T} = 0.8456$$

Now, the merged cost of both nodes is calculated by the two previous approaches, considering that:

$$c = cost_i^{S(i)} + (1 - \rho_{j,i}) \cdot cost_j^{S(j)}$$

$$\rho_{j,i} = (1 - \sigma_{i,j}) \cdot \kappa_{[j,S(j)],[i,S(i)]}$$

Therefore the costs by both methods can be easily found out:

$c_{standard}$ = 252 + (1 - 0.3281) · 200 = 387

$c_{macroscopic}$ = 252 + (1 - 0.1351) · 200 = 425

The *real* cost, considering the inner details of scheduling and allocation associated to the functional units, has been found to be 427, which is closer to the macroscopic approach than to the classical one.

Now, all the previous data have been calculated for *all* the possible degrees of parallelism between both nodes:

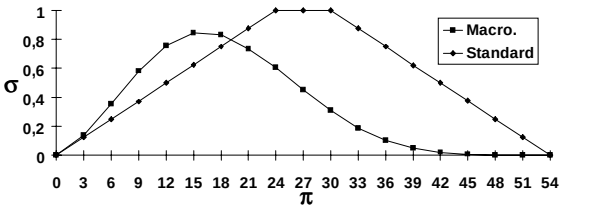

Figure 4

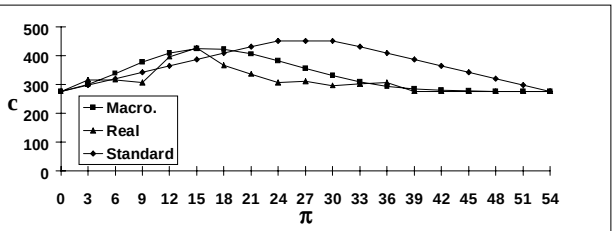

Figure 5

In Figure 4, the representations of the overlapping degrees found out by the standard and the macroscopic approaches are depicted. The costs calculated by both methods, as well as the *real* cost, are shown in Figure 5. The macroscopic approach obtains not only more accuracy, but also a higher fidelity degree.

5.- Conclusions and future work.

In this paper, an extension to the macroscopic approach has been presented, with the inclusion of a density parameter. Current efforts are being dedicated to the study of a vectorial approach, in which a separate density function is used for every kind of functional unit.

6.- References.

[1] D.D. Gajski et al., "Specification and Design of Embedded Systems", Ed. Prentice-Hall, 1994.

[2] J.A. Maestro et al., "A Macroscopic Time and Cost Estimation Model Allowing Task Parallelism and Hardware Sharing for the Codesign Partitioning Process", DATE'98.

[3] J. Henkel et al., "Adaptation of Partitioning and High-Level Synthesis in Hardware/Software Cosynthesis", ICCAD' 94.

[4] R. Niemann, P. Marwedel, "Hardware/Software Partitioning using Integer Programming", ED&TC'96.