Grey Relational Hardware-Software Partitioning for Embedded Multiprocessor FPGA Systems

1Yang-Hsin Fan, 2Trong-Yen Lee

1Dept. of Computer Science and Information Engineering
National Taitung University, yhfan@nttu.edu.tw
2Graduate Institute of Computer and Communication
National Taipei University of Technology, tylee@ntut.edu.tw
doi:10.4156/aiss.vol3. issue3.05

Abstract

Hardware-software partitioning in an embedded multiprocessor field programmable gate array (FPGA) system is difficult as such systems are uncertain and constraints are various. Moreover, the effect of relational degree for each partitioning combination of system constraints is too difficult analysis to determine a partitioning result. This work applies grey relational analysis to identify a partitioning result for an uncertain system with multiple constraints. Moreover, the relational effect of each partitioning combination is applied to determine the role of each task as hardware or software. Experimental results indicate that the proposed attains a partitioning result with low power consumption and fast execution time for a benchmark with 199 tasks.

Keywords. Grey Relational Analysis, Hardware-software Partitioning, Embedded Multiprocessor FPGA System, Hardware-software Codesign.

1. Introduction

Many electronics products are developed on an embedded multiprocessor field programmable gate array (FPGA) systems, which have the following advantages. First, flexible hardware and software architectures provide various computing abilities that meet performance requirements. Second, floating operation is more easily implemented by running applications on a processor platform than on a hardware circuit. Third, various functions associated with hardware and software tasks can be evaluated rapidly. Finally, the time and money required for system implementation are reduced compared to those for application specific integrated circuits (ASICs). Consequently, embedded multiprocessor FPGA systems have been adopted by both industry and academia. Industrial applications include the design and verification of portable devices, consumer electronics and telematics. In term of academic research, the fields are hardware-software partitioning, hardware-software codesign and hardware-software cosynthesis.

Hardware-software partitioning is a nondeterministic polynomial (NP)-hard problem that the problem leads to partitioning result that meets all system constraints and an optimal feasible solution too difficult to obtain. Therefore, many researchers [1]-[13] have developed various hardware-software partitioning schemes for effective partitioning. These techniques include genetic algorithm (GA) [1][3], multi-level partitioning (MLP) [4][5], recursive spectral bisection (RSB) and scheduling algorithm [6], hardware-oriented partitioning (HOP) [7], enhancement partitioning [8], efficiently partitioning [9], sophisticatedly computed partitioning [10] and Kernighan and Lin [13] partitioning. However, these studies did not discuss the effect of relational degree for each partitioning combination on system constraints in embedded multiprocessor FPGA systems. Consequently, this work presents a grey relational hardware-software partition scheme for obtaining a partitioning result with low power consumption and fast execution time.

The remainder of this paper is organized as follows. Background and preliminary work for hardware-software partitioning is introduced in Section 2. Section 3 addresses the grey relational hardware-software partitioning approach. Experimental results for a benchmark case with 199 tasks, are presented in Section 4. Finally, conclusions are given in Section 5.
2. Related work

2.1. Background

Genetic algorithm (GA) evolves successive generations via mutation and crossover to solve hardware-software partitioning problem. Saha et al. [1] used GA to transfer initially hardware-software partition into constraint satisfaction problem (CSP), which they solved to obtain the partitioning results in terms of cost, execution time and concurrency constraints. In 2010, Aoshima and Kanasugi [2] adopt redundant binary number in genetic algorithm for processor that can find the optimized solution and decrease error rates. Zou et al. [3] considered data communication time between hardware and software in a system when it performing hardware-software partition. Moreover, they also suggested that executing mutation in GA should depend on the fitness function rather than the mutation rate.


2.2. Preliminary work

A task is an atomic unit in an embedded multiprocessor FPGA system. Each task can be implemented as a hardware component or software procedure when developing an embedded multiprocessor FPGA system. In other words, each task has two types of implementation. The hardware-software partitioning technique becomes a decision system that determines each task to be implemented as hardware or software. Thus, an embedded multiprocessor FPGA system consists of a set of tasks that will be partitioned into two sets of hardware, set H and one software set, set S, after hardware-software partitioning.

This work developed a partitioning tool shown in Figure 1-2 which based on the Kernighan and Lin [13] algorithm. The partitioning tool can construct a bi-sectioned and balanced system of sets H and S. Additionally, the developed tool can reduce external cost via swapping a subset from H and with one from S. Figure 1 shows a case with 30 randomly generated weighted tasks (i.e., Nos. 0-29) comprised of H (left) and S (right) sets. This case has two costs: internal and external costs. An arc connects tasks within H or S; these arcs are called internal costs. External cost defines the cross connection between H and S tasks. This work discusses the un-weighted and weighted effect on internal and external costs in the developed tool. Figure 2 shows that the cost is reduced to 28.28% via swapping node (refer No. 11, 28 and No. 12, 16, etc.) from H and
with one from $S$. Although the ability of the developed tool was extended from the un-weighted effect to the weighted effect, the tool does not consider the effect of relationship degree for each constraint during swapping procedure.

**Figure 1.** A weighted bi-section and balanced system for randomly generated 30 tasks.

**Figure 2.** Reduced external cost for a 30 weighted tasks of bi-section and balanced system.
3. Grey relational hardware-software partitioning

3.1. Grey relational analysis

The grey system was developed by Deng [14] in 1989 to solve uncertain system problems. Equations (1)–(7) are typically applied in grey relational analysis when analyzing the grey relational effect. This work applies grey relational analysis to achieve grey relational hardware-software partitioning. 1) An embedded multiprocessor FPGA system can be partitioned into a set of tasks mapped to a set of $X_i(k)$, where $i$ is a hardware and software task and $k$ is a set of constraints. Therefore, an oriented dataset, $X_i(k)$, is generated by Eq. (1). 2) System constraints are set to $X_0(k)$, which is shown in Eq. (2.1). 3) Hardware set, $H$, and software set, $S$, which are shown in Eqs. (2.2) and (2.3), respectively, can be distinguished from $X_i(k)$. Hardware set $H$ is set to $X_{1hw}(k), X_{2hw}(k), \ldots, X_{mhw}(k)$, and software set $S$ is set to $X_{1sw}(k), X_{2sw}(k), \ldots, X_{msw}(k)$. 4) According to Eq. (3), $X_{ihw}(k)$ or $X_{isw}(k)$ becomes a candidate task, as task ${\text{candidate}}_{e}$ and task ${\text{candidate}}_{p}$ are determined by the largest number for computing $\gamma(X_0(k), X_{ihw}(k))$ and $\gamma(X_0(k), X_{isw}(k))$. Thus, each task can be identified and implemented as hardware or software. This procedure is repeated until $X_mhw(k)$ or $X_msw(k)$ is derived. Therefore, two partitioning results, comprising a set with task ${\text{candidate}}_{e}$ and task ${\text{candidate}}_{p}$, are generated. 5) A grey relational hardware-software partitioning result can be achieved by calculating the largest number of $\gamma(X_0, X_i)$ for the two partitioning results derived by Eq. (7).

\[
X_i(k) = (X_i(1), X_i(2), \ldots, X_i(k)) \in X \quad i=0, 1, \ldots, m \text{ and } k=1, \ldots, N
\]  

\[
X_0(k) = (X_0(1), X_0(2), \ldots, X_0(k))
\]  

\[
X_{1hw}(k) = (X_{1hw}(1), X_{1hw}(2), \ldots, X_{1hw}(k))
\]  

\[
X_{2hw}(k) = (X_{2hw}(1), X_{2hw}(2), \ldots, X_{2hw}(k))
\]  

\[
X_{3hw}(k) = (X_{3hw}(1), X_{3hw}(2), \ldots, X_{3hw}(k))
\]  

\[
\vdots
\]  

\[
X_{mhw}(k) = (X_{mhw}(1), X_{mhw}(2), \ldots, X_{mhw}(k))
\]  

\[
X_{1sw}(k) = (X_{1sw}(1), X_{1sw}(2), \ldots, X_{1sw}(k))
\]  

\[
X_{2sw}(k) = (X_{2sw}(1), X_{2sw}(2), \ldots, X_{2sw}(k))
\]  

\[
X_{3sw}(k) = (X_{3sw}(1), X_{3sw}(2), \ldots, X_{3sw}(k))
\]  

\[
\vdots
\]  

\[
X_{msw}(k) = (X_{msw}(1), X_{msw}(2), \ldots, X_{msw}(k))
\]  

\[
\gamma(X_i(k), X_j(k)) = \frac{\Delta \min + \Delta \max}{\Delta \max}
\]  

\[
\Delta_j(k) = |X_i(k) - X_j(k)|
\]  

\[
\Delta \min = \forall \ j \in i, \forall \ k \left| X_i(k) - X_j(k) \right|
\]
Grey Relational Hardware-Software Partitioning for Embedded Multiprocessor FPGA Systems  
Yang-Hsin Fan, Trong-Yen Lee  
Advances in Information Sciences and Service Sciences. Volume 3, Number 3, April 2011

\[ \Delta_{\text{max}} = \max_{j \in i, k} \max |X_i(k) - X_j(k)| \] (6)

\[ \gamma(X_i, X_j) = \frac{1}{n} \sum_{k=1}^{n} \gamma(X_i(k), X_j(k)) \] (7)

An example of power consumption during grey relational analysis in hardware-software partitioning is as follows. Suppose \( X_i \) is 181.35; \( X_j^{\text{hw}} \) and \( X_j^{\text{sw}} \) are then 0.383 and 1.067, respectively. The \( \Delta \) of \( X_j^{\text{hw}} \) and \( X_j^{\text{sw}} \) can be derived separately as 180.97 and 180.28 by Eq. (4). Next, \( \Delta_{\text{min}} \) and \( \Delta_{\text{max}} \), as shown in Eqs. (5) and (6), can be computed as 0.0027 and 2.92 from a dataset. The distinguishing coefficient \( \delta \) is typically set at 0.5. After these parameters of grey relational coefficients are calculated, \( \gamma^{\text{hw}}(X_i, X_j^{\text{hw}}) \) and \( \gamma^{\text{sw}}(X_i, X_j^{\text{sw}}) \) are derived as 0.0161 and 0.0162, as shown in Eq. (3). Thus, the candidate of task is to be implemented by software called task\(_p^{\text{candidate}}\) as \( \gamma^{\text{sw}}(X_i, X_j^{\text{sw}}) \) is larger than \( \gamma^{\text{hw}}(X_i, X_j^{\text{hw}}) \).

### 3.2. Grey relational partitioning algorithm

The grey relational partitioning algorithm, GreyRelationalHW-SWPartition() (Fig. 3), can obtain a partitioning result with low power consumption and rapid execution time. The algorithm requires five input elements—\( X_0, X_e^{\text{hw}}, X_p^{\text{hw}}, X_e^{\text{sw}} \) and \( X_p^{\text{sw}} \). System constraints are set to \( X_0, X_e^{\text{hw}} \) and \( X_p^{\text{hw}} \), which represent execution time and power consumption of the hardware task, respectively. Additionally, \( X_e^{\text{sw}} \) and \( X_p^{\text{sw}} \) are execution time and power consumption of the software task, respectively. Each task will be implemented as hardware or software after computing the grey relational value of \( \gamma_e^{\text{hw}}, \gamma_e^{\text{sw}}, \gamma_p^{\text{hw}} \) and \( \gamma_p^{\text{sw}} \) for identifying candidate tasks. After all candidate tasks are identified, two partitioning results, task\(_e^{\text{candidate}}\) and task\(_p^{\text{candidate}}\), are generated. Finally, grey relational analysis is adopted to obtain a partition result.

Figure 3 shows the proposed GreyRelationalHW-SWPartition() algorithm for grey relational hardware-software partitioning. The running time of GreyRelationalHW-SWPartition() is \( O(n) \) for labels 1–12. Label 13, \( O(m\log m) \), is used to find the maximum grey relational value for task\(_e^{\text{candidate}}\) or task\(_p^{\text{candidate}}\). Therefore, the time complexity of GreyRelationalHW-SWPartition() is \( O(n^2m\log m) \).
Algorithm GreyRelationalHW-SWPartiton($X_0$, $X_e^{hw}$, $X_e^{sw}$, $X_p^{hw}$, $X_p^{sw}$)

**Input:** $X_0$ is system constratins of execution time and power consumption, $X_e^{hw}$ and $X_e^{sw}$ are execution time and power consumption of hardware tasks. $X_p^{hw}$, $X_p^{sw}$ are execution time and power consumption of hardware tasks.

**Output:** One partitioning result

{ for (each task) do
  
  Computer grey relational value of $\gamma_e^{hw}$, $\gamma_e^{sw}$, $\gamma_p^{hw}$ and $\gamma_p^{sw}$
  
  If ($\gamma_e^{hw} > \gamma_e^{sw}$) then
    { task$_e$ candidate = task$_e$ candidate & task$_e^{hw}$ }
  else
    { task$_e$ candidate = task$_e$ candidate & task$_e^{sw}$ }
  
  If ($\gamma_p^{hw} > \gamma_p^{sw}$) then
    { task$_p$ candidate = task$_p$ candidate & task$_p^{hw}$ }
  else
    { task$_p$ candidate = task$_p$ candidate & task$_p^{sw}$ }

  Computing and find maximum grey relational value of task$_e$ candidate and task$_p$ candidate as partitioning result
}

**Figure 3.** Grey relational hardware-software partitioning algorithm

4. Experimental results

This experiment makes eight assumptions for focusing on the discussion of grey relational hardware-software partitioning. First, we assume any embedded multiprocessor FPGA system can be divided into a set of tasks that can be modeled by a task graph. Second, all tasks can be implemented by hardware and software. Third, the constraints of power consumption and execution time for each hardware task and software task can be measured. Fourth, hardware tasks consume more power consumption than software tasks. Fifth, hardware tasks have faster execution time than software tasks. Sixth, implementing a hardware task only requires slices of FPGA, as opposed to software tasks the mere needs memory of FPGA. Seventh, interface cost is zero. Finally, every processor in a multiprocessor has the same characteristics implying that each processor has same computation for each software task.

This work uses a 199-tasks benchmark [15] to demonstrate the feasibility of the proposed scheme. As a well-defined benchmark is not supplied by academia or industry, Purnaprajna [15] et al. presented a case with 199 tasks and a task graph generated randomly to simulate an embedded system. This study implements a graphical user interface that generates a task graph randomly (Figure 4). The evaluation parameters include execution time and power consumption. The execution time parameters for hardware and software tasks, 0–4 and 0–30, respectively, were generated randomly. The power-consumption parameter for hardware and software tasks was generated randomly, and was 0–3 and 0–1. The distinguishing coefficient $\delta$ of grey relational coefficient is typically set to 0.5. This experiment was performed five times (column 1 in Table 1). Columns 2 and 3 in Table 1 lists experimental results obtained using the proposed method. Columns 4 and 5 in Table 1 show the case 1 results of obtained by Purnaprajna [15]. Columns 2 and 4 in Table 1 show comparison results for power consumption, indicating that the proposed method has low power dissipation for all cases. In terms of execution time
(columns 3 and 5 in Table 1), experimental results indicate that the proposed method for each case has fast execution time.

![Figure 4. An embedded system of simulation with 199 tasks.](image)

**Table 1. Comparison of the proposed method and Purnaprajna [15]**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Power</td>
<td>Time</td>
</tr>
<tr>
<td>Task 199-1</td>
<td>104.99</td>
<td>420.60</td>
</tr>
<tr>
<td>Task 199-2</td>
<td>104.99</td>
<td>420.60</td>
</tr>
<tr>
<td>Task 199-3</td>
<td>276.98</td>
<td>291.32</td>
</tr>
<tr>
<td>Task 199-4</td>
<td>269.83</td>
<td>277.15</td>
</tr>
<tr>
<td>Task 199-5</td>
<td>104.99</td>
<td>420.60</td>
</tr>
</tbody>
</table>

5. Conclusions

Hardware-software partitioning is used to solve a partitioning problem for embedded multiprocessor FPGA systems. This work applies grey relational hardware-software partitioning to obtain a partitioning result. Moreover, grey relational hardware-software partitioning algorithm with a time complexity of \(O(n^2 \cdot \log m)\) is presented. Specifically, the proposed method achieves a partitioning result with low power consumption and fast execution time via grey relational analysis. Experimental results of benchmark with a 199 tasks demonstrates the feasibility of the proposed approach for hardware-software partitioning in embedded multiprocessor FPGA systems.
6. Acknowledgments

The authors would like to thank the National Science Council of the Republic of China, Taiwan, for financially supporting this research under Contract No. NSC-95-2221-E-027-131.

7. References