Intellectual Property (IP) Integration Approach for Data-Flow Parallel Embedded Systems

Anne Marie Chana\(^1\) and Patrice Quinton\(^2\)

\(^1\) National High School Polytechnic, University of Yaounde I, Cameroon
anne.chana@yahoo.fr

\(^2\) ENS Cachan Bretagne, Université européenne de Bretagne Campus de Ker Lann,
35 170 Bruz, France
patrice.quinton@bretagne.ens-cachan.fr

Abstract. The growing complexity of new chips and the time to market constraints require fundamental changes in the systems design approach. Systems on chip (SoC) based on reused components called intellectual property (IP) has become an absolute necessity to the embedded systems companies in order to remain competitive. This paper focuses on the IP reuse to design parallel and multi-frequency applications. The flexible parallel components described by the ALPHA functional language are modelled and assembled using a scheduling method which combines the synchronous data-flow principle of balance equations and, the polyhedral scheduling technique. Our approach allows a flexible component to be modelled and, a full system to be assembled and synthesized by combining the component hardware descriptions with automatically generated wrappers. We discuss the relationship of this approach with stream languages, latency-insensitive design, and multidimensional data-flow systems.

Keywords: IP reused, flexible component, SoC, Polyhedral model, data-flow model, parallelism, multi-frequency system.

1 Introduction

Over the past 20 years, embedded systems have become the supports of the most advanced hardware and software technologies. In this period, the implementation of embedded system technology has evolved from microcontroller to fully integrated systems-on-chip (SoC). SoCs and related technologies are driving embedded systems today and in the foreseeable future. New demanding applications in terms of processing power appeared over the past 20 years, driven by PC boom, Internet and wireless environments. These applications are called systematic and intensive processing and are found both in scientific computing and signal processing (telecommunications, multimedia processing, image processing and video, etc.) for exemple, in short in cyber-physical \(^1\) systems in general. Processing such applications requires systematic and extensive high
capacity data processing often using techniques of parallel and distributed computing. The difficulties to develop those applications are primarily due to the exploitation of data and arithmetic parallelism, time and resources constraints.

Many such systems are designed by re-using existing software or hardware modules often called flexible blocks or Intellectual Property (IP). These flexible components are available as software or hardware functions and represent specific application elements for signal processing blocks (DCT, FFT, etc.), telecommunications (Viterbi codes, Turbo-codes, etc.), or Multimedia (MPEG2, MPEG4, JPEG, etc.).

In practice, it is not easy to assemble and operate components from different designers. Even if they are tested beforehand, there is no guarantee that, when put together the system will work correctly.

Integrating components together must take into account several aspects. The components must be synchronized to allow a correct overall operation of the system, and to ensure proper data exchange and valid communication protocols. Inputs/outputs must sometimes be stored to meet the synchronization constraints.

We present in our paper a systematic method to automatically generate a hardware architecture for multi-frequency, parallel data-flow (or streaming) systems with flexible generic components. The flexible components that we use are similar to those synthesized using the MMALPHA environment, which allows parallelization and hardware synthesis. They can model parallel architectures operating on multi-dimensional streams of data, in a regular fashion.

Our method consists of specifying such flexible components, their scheduling and synthesis, and then determines their assembly conditions. If the assembly is possible, we produce a hardware architecture by adding a wrapper to each component so that the whole system becomes synthesizable. The wrapper is used to control the frequency utilization of the component and achieve the system control.

Our scheduling technique combines the method used in the synchronous data-flow model [5] and the polyhedral model method, as described in [6]. This schedule allows one to deduce the different logic sub-clocks that meet the I/O constraints of each component, and the number of additional registers that may be needed between some of the components.

In this paper we present the main ideas of our method; our main contribution is to show how automatic method allows one to synthesize such complex system. We present the steps of the automatical synthesis that starts with the description of the system by ALPHA and ends with the scheduling and automatic generation of synthesizable code.

2 Integration Approach

Our approach to integration consists of several step, namely:

- The system specification: behavioural description of the system as SDF graph and checking the system hardware feasibility. In this step we also select the suitable behavioural flexible components (IPs) to use.