COMPSYS701 Advanced Digital Design, 2022

Individual Research Project (40%)

Zoran Salcic and Morteza Biglari-Abhari

The Individual Research Project (IRP) is related to the research of functionalities that are primary candidates for hardware acceleration of digital signal processing algorithms that will be used as Application Specific processors (ASP) in Heterogeneous Multiprocessor System-on-Chip (HMPSoC). It has the following requirements on the individual students:

1. Exploring typical functions (kernels) that are often used in fields such as digital signal processing (DSP) and artificial neural networks and identification of those functions that would be of interest for hardware implementation and encapsulation into an Application-Specific Processor (ASP) within the HMPSoC; this ASP will be referred to in the project as Data Processing ASP or DP-ASP.
2. Selection of functions (kernels) for implementation, with the examples of functions specified below, implemented in digital hardware as a part of DP-ASP; the final choice will be made as a proposal that has to be approved by teaching staff. These functions can be executed, which is initiated by calls from ReCOP, or configured for automatic chaining and execution on data coming from the ASPs connected to TDMA-MIN NoC in critical part of HMPSoC.
3. Defining the architecture of a DP-ASP that implements those selected functions within your DP-ASP; this must include specification of an interface of ASP with the TDMA-MIN NoC, as well as the protocol how the functions are used.

Application-Specific Processor

As suggested from previous research, the DP-ASP has to be able to efficiently perform multiply-and-accumulate (MAC) operation on the requested number of elements of two arrays, A and B, stored in two physical memory blocks of 512 24-bit words considered to be signed integers, or on the stream of inputs delivered from other ASPs. This is useful for implementation of scalar products of two vectors represented by two arrays, different types of digital filters, correlation function etc. The data stored in these arrays come from either external world through a specialised IO-ASP, from other DP-ASP or another GPP (e.g. Nios II) via NoC.

Examples of connections to the external world are Analogue-to-Digital Converters (ADC) with associated ADC-ASP, Digital-to-Analogue Converter (DAC) with associated DAC-ASP, and similar sources/destinations that can be connected to the ASP directly or encapsulated into specialised ASPs connected to the common NoC.

The examples of kernel operations of a DP-ASP are:

(1) Direct passthrough, taking input data samples and immediately outputting them to a specified destination over NoC
(2) Linear Filter - moving average filter on any array A or B, storage of the resulting array back to the original location or delivering it to another ASP. Moving window size is programmable and can be L= 4 or L=8. For the boundaries of array, a solution has to be defined.

$$Y(i) = \frac{\sum \mathrm{X(k)}}{\mathrm{L}}, k = i, i+1, \dots, i+L-1$$

where X is either A or B.

(3) Finite Impulse Response filter of Nth order with fixed coefficients
(4) Correlation function (between two different time series stored in two arrays or over the same time series – autocorrelation)
(5) Peak detection where the ASP records data to the internal memory and outputs the current maximum and minimum values for the specified/sampling period
(6) Averaging – where the ASP computes a rolling average over the specified/sampling period
(7) Frequency detection of the incoming data received via NoC
(8) Accelerating machine learning (ML) computations such as computations in convolutional layers of a Convolutional Neural Networks (CNN)

As discussed in the project brief, ASPs will be configured using instructions from the ReCOP and/or GPP if relevant (see also: "Project Technical Notes", will be provided on Canvas) and will operate autonomously to perform their configured task, and may be configured into a pipeline of ASPs to achieve the application requirements. You may extend the ASP with additional functions that are useful to the tasks of data processing.

As an example, one configuration might involve three DP-ASPs, where ADC channel 1 being passed through a linear filter, then to peak detection, with the peak detection output to the DAC channels, leading to the following configuration:

1. The DP-ASP used for peak detection should be configured to peak detect mode, with output set to the DAC-ASP address
2. The DP-ASP used for linear filtering should be configured to linear filter mode, with the output set to the DP-ASP used for peak detection
3. The ADC-ASP should be configured with the desired channel to output to the ASP performing linear filtering

Once this configuration is complete, the ADC-DSP will periodically output data which will autonomously be passed through the required pipeline of DP-ASPs to the DAC-ASP. Prior to reconfiguration, the ADC-DSP should be deactivated to avoid spurious data travelling through the datapath.

Limited testbenches will be provided to demonstrate core ASP functionality, you will be expected to expand on these to fully exercise the functionality of your DP-ASP.

In order to design the DP-ASP, all command responses of the DP-ASP have to be precisely defined and format of all packets exchanged between nodes on NoC (ReCOPs and ASPs) be specified. The RTL level model of the DP-ASP has to be defined and it has to include all data storage elements, interface to the NoC interface (NI), datapath of ASP and its control unit. Analysis of performance/area trade-off (time to perform the operation against the required resources to implement the DP-ASP) for the ASP should be performed and the options of parallelisation of the required operations be explored, which can be achieved by using more processing elements such as multipliers, adders etc within the datapath.

In the adopted approach to critical part of HMPSoC as specified in the design requirements documents, network interfaces for all types of nodes are identical.

In some application scenarios, DP-ASPs have to provide internal memory for temporary storage of data, such as previously described memory blocks (A and B) for storage of arrays. Access to these memories is controlled by dedicated controllers that enable, for example, continuous storage of a part of a stream of data of certain length in real-time and making it available for processing within DP-ASPs.

In this work we will assume that a maximum number of nodes connected to the single TDMA-MIN is 8 and the datapath within the NoC fabric allows transfers of 32 bits from any node to any other node at once (in a single clock cycle).

The result of the research has to be presented in the following forms:

- 6 to 8-page single column report (10 pt. font, Times New Roman) that includes the descriptions of findings on related topics in the literature, short presentation/definition of implemented algorithms, diagrams of overall DP-ASP structure and its datapath, summary of capabilities and the ways of using the DP-ASP, and references.
- The design of the DP-ASP; individual designs will be used by the design team to integrate into HMPSoC. The design has to be tested with realistic external requirements implemented within the testbed (in ModelSim) and synthesised into an IP block that can be instantiated in HMPSoC in any number of instances in DE1-SoC (or DE2-115).

More details on IRP and expectation from individual students will be given in lecture time and in direct consultations.