Individual Project Report

Peak Detection Application-Specific Processor

Brian Wei

ABSTRACT

This individual project documents the development of a Peak Detector Application-Specific Processor (PD-ASP) designed as one of the four Data Processing Application-Specific Processor (DP-ASP) to be integrated together on a Heterogeneous Multiprocessor System-on-Chip (HMPSoC) for hardware acceleration of digital signal processing algorithms. This will be applied in a simplified real life application in the form of a frequency measurement of a periodic signal.

It utilises the instructions from ReCOP (Reactive and Concurrency Oriented Processor) and/or Nios II processor to configure the PD-ASP and ensures that it can run autonomously to perform its configured task of detecting peaks in the signal.

The report will detail the design process from start to finish, showing the modifications and optimizations that are made along the way to ensure that the design specifications are met as well as improving the system's performance overall with other DP-ASPs, ReCOP, Nios II, and HMPSoC.

Keywords: Nios II, VHDL, Digital System design, ASP, Frequency measurement, TDMA-MIN NoC, HMPSoC, PD-ASP.

1. Introduction

The Individual Project (IP) of COMPSYS 701 involves the design and development of an Application-Specific Processor (ASP) where it will contain specific functionalities as determined by a team of four people to ensure the hardware acceleration of digital signal processing algorithms. Each person is to be tasked with designing and developing one of the determined functionalities. The IP is the second part of a bigger project which by the third part will combine the four developed ASP, each with their respective unique functionalities and the Reactive and Concurrency Oriented Processor (ReCOP) to form a Heterogeneous Multiprocessor System-on-Chip (HMPSoC).

The ASP discussed in this report will involve the Peak Detection functionality which will be referred as PD-ASP from here on. The report will touch on the goals of developing the PD-ASP, its relevance to the overall integration with other ASPs and a discussion on design choices that are made throughout the development and how it was implemented.

2. Project objectives

The primary objective of the IP is to develop an ASP that will take care of one of the four decided functionalities to be implemented. In this case, it will focus on developing an ASP that detects the peak of the signal and having its result be forwarded to Nios II for further processing. The PD-ASP should have precisely defined formats for the exchanging of packets between the nodes on NoC (ReCOP, NiOS II and ASPs).

2. PD-ASP

The PD-ASP is designed so that it complements easily when integrating with the other ASPs, Nios II and ReCOP. With this goal in mind, determining how the data will be transmitted and exchanged between the components above will be a crucial aspect of the development. Also due to the numerous conditions that are needed for determining where the peak is, a finite state machine (FSM) is included into the design for better memory management throughout the operation of the PD-ASP.
After consulting with the developer of the Correlation Application-Specific Processor (COR-ASP), the format of the data exchanged between the two is made. It will consist of two parts: the address packet and the data packet. It is decided that for the address packet, its size will be 8 bits while the data packet will be 32 bits. The address packet is used to direct which node to go in the TDMA-MIN NoC. For the data packet, it is decided that the 31st bit will be

used as a validation bit which notifies to the PD-ASP that the incoming packet is valid, then the 30th bit is used as a correlation ready bit. This bit is set to high when the packet contains a determined correlation value from the COR-ASP. The remaining bits will contain the calculated correlation value from COR-ASP itself.

Methodology

The PD-ASP can receive packs from two sources: the COR-ASP and from the Nios II processor. The Nios II processor will send configuration packets to the PD-ASP which configures the PD-ASP to be enabled and directs where the output should go. For the COR-ASP, it will receive packets containing the 20 bit correlation value as explained above. It will be stored in one of two temporary buffers where one is used to store the current correlation value received while the other stores the previous correlation value. When both buffers are non-zero, meaning they both have correlation values, it will enter the start state of the FSM. In this state, it determines whether the starting point of the analysis of the signal is a rising slope or a falling slope. For it to be rising, the current correlation value must be larger than the previous correlation value. If the current correlation value is smaller than the previous correlation value, then the starting point is a falling slope. Each of these conditions will transition to their respective states, being either rising state or the falling state, while also incrementing a counter called state_counter by one which keeps track of how many states it has transitioned from the start state. Both start and falling states have very similar logic where if it encounters the current correlation value is either equal or greater than or less than, it will stay in their respective states. With each comparison, a separate counter will be incremented by one to keep track of the time taken to detect the peak. However, when it detects that the opposite conditions for what makes it rising state or falling state respectively, then it will switch to the other state, so if originally it was in rising state, it will switch to falling and originally in falling state, it will switch to rising state. When it detects that the state_counter has the value of 3, in whatever state it is in, it will set up, configure and send the accumulated counter value to the Nios. While this is happening, it reset the state_counter and the counter back to 0 to restart the process of detecting when the next peak is found. All of this happens every one clock cycle.