COMP4211 Advanced Architectures & Algorithms
Project Proposal

Main Reference:
A New Look at Exploiting Data Parallelism in Embedded Systems
Hillary Hunter, Jaime H. Moreno (IBM eLite DSP team)
http://www.research.ibm.co/elite/Publications/hunter_cases03.pdf

Motivation
SIMD architectures were originally motivated by throughput needs of scientific, military & air traffic applications. SIMD extensions were added to desktop processors in 1990s for gaming and graphics processing. Today, embedded applications (mostly telecommunication and media applications) exhibit repetitive computations across large data sets. High level of data-level parallelism (DLP) can be extracted from these applications. Conventional SIMD (referred to as SIM_pD hereafter) exploit DLP by executing a single operation on multiple data elements packed into a single (vector) register. The packed data elements could potentially come from disjoint sources and may be written back to disjoint destinations. However, current implementations of SIM_pD do not allow for this level of data flexibility; creating a gap in the design space of embedded systems between VLIW and SIM_pD architectures (as shown in the figure below – extracted from the main reference paper). The IBM eLite DSP team investigated on how best to extend the SIM_pD architecture to allow for flexible data patterns using an indirect-SIM_dD implementation. The team also provided quantitative measures of how well telecommunication and media kernels inherently map to SIM_pD and SIM_pD patterns.

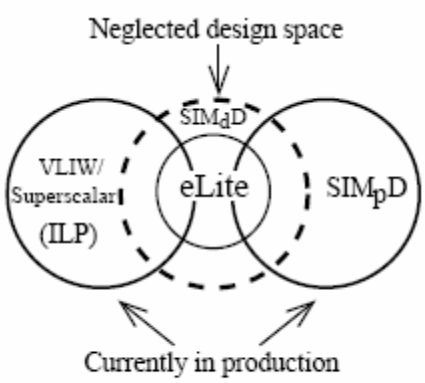

Figure 9: Partial architecture design space.

Description of Main Task
Based on a selected set of kernels commonly used in embedded (DSP) applications, investigate the potential of SIM_dD capabilities as an alternative to current embedded architectures such as the VLIW and SIM_pD architectures.

Breakdown of Tasks

Uni Week No.	***Tasks***
Week 7	• Literature review on VLIW, SIM_pD and SIM_dD architectures as described in the main reference paper.
Week 8	• Select a small set of commonly used DSP kernels • Obtain descriptions of these kernels in high-level language (C) and assembly language (MIPS-like). • Identify the set of instructions needed and modify the ISA of the basic 5-stage pipelined MIPS architecture (refer to Hennessy and Patterson's textbook)
Week 9 Week 10	• VHDL implementation of the basic MIPS architecture (modify the pipelined implementation from 02s2 COMP3211 assignment) • Extend the basic MIPS architecture to support VLIW, SIM_pD and SIM_dD extensions. Provide VHDL models of these extensions using Xilinx ISE.
Week 11	• Investigate effect of VLIW on selected kernels via functional simulation using ModelSim • Modify ISA?
Week 12	• Investigate effect of SIM_pD on selected kernels via functional simulation ModelSim • Modify ISA?
Week 13	• Investigate effect of SIM_dD on selected kernels via functional simulation ModelSim • Modify ISA?
Week 14	• Report write-up