Chapter 11: Conclusions and Suggestions for Future work

11.1 Conclusions

The research work examines the issue of leveraging ASIP architecture and describes the design methodology, instruction set design and the micro-architecture design approach for building an Application Specific Instruction set Processor (ASIP) for delivering greatly improved performance and reduced power consumption while maintaining flexibility in the implementation of applications. Designing appropriate architecture and micro-architecture are the keys to developing optimal solutions for promising new embedded applications. This work has looked at one such application – the unlimited vocabulary text-to-speech conversion in real time using formant based parametric speech synthesis.

By following a micro-architecture design approach that allocates computational load-driven, optimally-architected, dedicated functional units for each of the most frequently needed operations and functions, one is able to define ‘high-level’ application-specific, user-friendly instructions (besides the necessary low-level, general-purpose type instructions) and support their high-performance execution through a staggered launching of computational operations associated with a ‘high-level’ instruction via successive micro-instructions on a datapath that has only a single bus, but whose most important functional units take multiple clock cycles to complete their operations. Thus, the approach very effectively and efficiently reduces the ‘semantic gap’ between the application and the machine language instruction set of the processor.

It has also been established that this approach leads to a much lower power consumption for the application vis-à-vis its execution on a RISC processor. This is because the approach we have chosen minimizes the long distance data moves and instruction fetches over internal buses (that have large capacitances) in comparison to the RISC architecture – because RISC processors use simpler application-non-specific functional units, and therefore, require many more data transfers over the internal buses to implement the same application.
The net result is a very compact program code (that minimizes program memory size and instruction fetches), a rather limited micro-code, a reduced semantic gap, greatly reduced power consumption and improved performance for the application.

Introduction of application specificity in architecture and micro-architecture can deliver huge performance and power gains at a given technology level. Tools need to be developed to make ASIP design and application re-targeting on the ASIP as convenient and efficient as software code development for the ASIP approach to become popular.

### 11.2 Suggestions for Future Work

In order to make the ASIP approach more popular, design cycle time for ASIP development and application targeting/retargeting on the ASIP needs to be made much more fast. This can be achieved by developing a tool chain that supports a complete automation of the design approach explored and established in the thesis. The benefits of developing such a tool chain would be immense in terms of adoption of the ASIP approach by industry.

Based on our exploration of the formant based parametric speech synthesis approach as an instance case for proposing, developing and establishing the ASIP approach we can also say that real-time image processing and video processing required for smart cameras for various security and industrial applications is another very important application for exploring and developing ASIPs.

We do believe that architectural leveraging (whether as ASIPs or through reconfigurable computing or a mixture of both) holds an immense potential for delivering continuing performance improvement through chips even when the magic rug of Moore’s law gets pulled from under the feet of the chip designers. That this is perhaps one important direction to explore for delivering continuing performance enhancement in the post-Moore era.