**Compiler infrastructure to support power aware course grained reconfiguration in FPGA’s**

Main goal of the research – To provide compiler suite to allow for compile time application scheduling support for dynamically reconfigurable architectures. Large part of research will involve exploring topics along lines of adaptive application scheduling/architecture reconfiguration based upon power/performance characterization. The compiler will exploit the following –

By dynamically programming the board memory on the fly using fast on-chip configurable memory (MRAM based?), the compiler will make decisions as to how the code will be scheduled runtime for achieving optimal power/performance. At a very high level, the compiler will be helpful to configure number of cores the applications are running on and the network topology by fast reconfiguration of the architecture using the new memory architectures which provide support for fast reconfigurability. In addition, the compiler might also make decisions on code restructuring for optimal execution on course grained reconfigurable architectures.

Preliminary steps for understanding the course grained reconfigurable architectures and how to make decisions on such optimizations –

Steps involved –

1. **Understand how to use existing reconfigurable architecture board on which we are running applications and performance of those applications -**

Procedure –

1. With help of Xiaobin, understand how he is currently running applications on the reconfigurable architecture he has described. He is able to configure various cores statically and is able to run benchmarks and obtain power/performance metrics. It will be very beneficial for me to redo the process independently so that I get an understanding the concepts of dynamic reconfigurability.
2. Xiaobin has asked me to investigate and look into SPREE based processor generation and integration into existing board. His idea is that the cores will be more lightweight than existing NIOS based processors. I plan on doing this so that I can compare the power/performance tradeoffs between NIOS and SPREE based cores
3. A rudimentary compiler architecture developed on top of STREAMIT compiler (as this was mentioned in the proposal). The goal of this effort will be to study if I can dynamically modify the number of cores and do application scheduling on cores in such a way that only few cores will be active at certain point. While this does not encompass all the advantages of having fast memories etc. it will still give me a base level infrastructure to do more complex compiler as we move forward in the research.
4. **Understand the runtime memory reconfiguration and compiler infrastructure needed to support such reconfigurability**

Procedure –

1. Study the MRAM based architectures and integrate them into existing board?(not clear)
2. See if I can obtain an alternate board on which these memories are already supported like this one http://www.eetimes.com/document.asp?doc\_id=1305908? (not clear)
3. Enhance the compiler to reprogram MRAM to do dynamic reconfigurability of both cores and network architecture
4. **Start investigating compiler specific optimizations/scheduling constraints especially wrt application specific optimizations for power**

Procedure – I don’t yet know the architecture of the compiler and how it will pan out, but a lot will be dependent upon the tasks 1 and 2 and what I learn there. So will keep enhancing this section as I keep making progress on section 1 and 2.