Chapter 5

Conclusions and Future Scope

In this thesis, we have investigated some application that required intensive memory management to improve its performance while executing in real life environments. This thesis discussed a hardware approach that use by software for safe and secure manual memory management via low overhead, comprehensive use-after-free checking with disjoint metadata. To efficiently implement application using intrinsic proposed here, we used ideas from existing software-only approaches, explored ISA support for explicitly identifying pointer operations, described efficient decoupled register metadata via register renaming copy elimination, and utilized a lock location cache to accelerate checks. The resulting overhead is likely low enough to use in production (and not just debugging). The resulting system ensures full memory safety, provides comprehensive protection from use-after-free, and buffer overflow while application running and using memory block of large size. We introduced HAMM, a cooperative hardware-software technique that allows quick reallocation of memory blocks by finding unused memory block using instruction which in implemented in processor itself. HAMM provides new primitives for faster recollection of used memory in hardware without limiting the flexibility of software procedure and this is use to find such unused memory block. Our evaluation shows that such approach reduces acceptable amount execution time spent on memory management.

The thesis has addressed one of the remedy of Real Time Distributed application where intensive memory requirement while executing real time data set and producing output. The simulation results shows the reduction in processor cycle that is used in executing memory related instruction can give benefit to overall performance of application as well also speed up the processor.

Power consumption is the other key issue. For classical processor-based architectures, the main source of power consumption is due to memory access Measurements on a standard processor architecture show that the percentage of energy required by instruction and data memory are typically above 70% of the overall value, and that every tested algorithm roughly presents the same distribution. Since every clock cycle a new instruction is fetched, the only way to reduce instruction's memory energy consumption is to reduce the number of execution cycles and this proposed instruction doing in our customized processor that can be seen by comparing result of processor benchmark using Dhrystone in Figure 4.4 and Figure 4.5. In fact, instruction memory consumption scales proportionally with speedup. On the other hand, several tests showed on our customized processor that access to data memory roughly scales with speed-up that can be seen in Table 4.7 of Chapter 4 of this thesis too. In fact, the execution on customized processor with our proposed instruction and C intrinsic allows an improved efficiency for data management by managing memory block using our proposed algorithm in Chapter 2 of this thesis, which manage memory block efficiently and thus also reducing data memory consumption and reduce the unused memory block in real time application and speed up the application execution which can be seen in Figure 4.7 of chapter 4 in this thesis.

Following are the areas where further research can be carried out:

1. Investigation on new methodology that automates the selection of very complex processing instruction set extensions for special purpose real time application together with aggressive techniques to map the basic blocks to such complex instructions.

2. Investigation on a basic ASIP core to extend it automatically to include ad-hoc functional units that accelerate the dataflow sections of the software application.

3. Investigation on developing different application specific general-purpose processor that can solve other issue of proc ssing like improving real time response and self-learning user define complex operation without major customization.

4. Investigation on designing plug and play type of general-purpose processor, which can do user and application specific operation for real world portable electronic devises.

5. Investigation on how to find dataflow instructions with more than one output using optimal algorithm and represented by a set of polynomials for the symbolic mapping step.

6. Investigation on designing general purpose processor which can share the user define instruction with traditional processor and assist it in improving overall system performance for real time application.

7. Investigation on to find the hotspot in software module of the application that would be encode in processor to speed up the application performance and improve the software response.

8. Investigation on introducing instruction which can consider as multi operational instruction and do multiple task in single execution of processor cycle for real time application to reduce the line of code.