Simultaneous thin-thread processors for low-power embedded systems

Won W. Ro\textsuperscript{1a)}, Jaeyoung Yi\textsuperscript{1}, Joon-Sang Park\textsuperscript{2},
and Joonseok Park\textsuperscript{3}

\textsuperscript{1} School of Electrical and Electronic Engineering, Yonsei University, Seoul, Korea
\textsuperscript{2} Department of Computer Engineering, Hongik University, Seoul, Korea
\textsuperscript{3} Department of Computer Engineering, Inha University, Inchon, Korea
\textsuperscript{a)} wro@yonsei.ac.kr

Abstract: The traditional pipelined RISC processors have been the mainstream technology in high-end embedded systems; it is due to that embedded applications are often satisfied with minimum performance requirement. However, the upcoming application domain of embedded systems will require more advanced microprocessor cores due to the future computing demands. Therefore, the system would eventually have to turn to advanced processors such as superscalars for the embedded processing cores. A drawback is that the conventional design of the superscalar processors possesses inherent complexity and power problems which are not easily acceptable in the domain of embedded processors. In this paper, we investigate the possibility to use multi-threaded processors to solve the problems with the traditional superscalar processors in embedded systems.

Keywords: embedded systems, ILP, TLP, superscalar, multithreading

Classification: Science and engineering for electronics

References


1 Introduction

Fast growing computing technology and the Internet service have dramatically changed our everyday life. In the last decade, countless new products have been rapidly developed and introduced to the consumers, including portable media players, high-performance digital cameras, high definition televisions (HDTV’s), and DMB (Digital Multimedia Broadcast) enabled cellular phones or PDAs (Personal Digital Assistants). These products are continuously demanding high-performance computing power and require more powerful embedded processors inside; they need more sophisticated functional features such as high-quality video and audio processing, high-speed data transfer, security issues, and gaming performance.

The microprocessor architectures in embedded systems have constantly advanced following the historical evolution of desktop microprocessors. In fact, the majority of recent commercial embedded processors is based on the early RISC model and pipelined architectures. Due to the typical design criterion of embedded applications – the designers normally target to achieve the required performance at a minimum price rather than pursuing higher performance, the commercial market has not felt the need for superscalar embedded processors yet. However, as mentioned earlier, many commercial products and portable devices would need more computation power in the near future, and consequently they would require more advanced embedded microprocessors.

The superscalar architecture can provide the high performance computing power that the future embedded systems will require. However, using superscalar processors in embedded systems contains one big challenge. The dynamic scheduling of the superscalar model consumes tremendous power and contains inbuilt circuit complexity. It is mainly due to that the current superscalar architecture is fundamentally utilizing the hardware scheduling logic to achieve multiple instruction issue and out-of-order execution. Future superscalar models are aimed to support more in-flight instructions, more issue-width, and aggressive speculation. It will correspondingly spend more and more power, making the future wide issue superscalars inadequate for the future embedded processors.

In this paper, we develop and propose a low-power multithreaded processor model for future embedded systems. The proposed architecture model
STT (Simultaneous Thin-Threading) has been developed based on the existing SMT (Simultaneous Multi-Threading) architecture; however, instead of implanting a large issue queue and a wide issue width, our STT model minimizes the size of hardware logic and dynamic scheduling capability. In other words, STT is designed to provide low power architecture by limiting the instruction parallelism inside a single thread. However, it still adopts the multithreading capability of SMT architecture to exploit ILP between threads.

2 Problems using superscalars in embedded systems

Although forecasters have predicted that the microprocessors in the future embedded system would adopt superscalar techniques for high-performance, only a limited number of products have been introduced in the commercial market; it is mainly because of the complexity of the superscalar architecture. Superscalar strongly depends on the hardware scheduling logic to achieve multiple instruction issue and out-of-order execution. Therefore, the single issue pipelined processors (also known as scalar processors opposed to superscalar ones) have been successful and noteworthy in the commercial market; the computing demands of embedded applications have been fulfilled well with the lower-price processors.

In addition, the power consumption level of single issue scalar processors is superior compared to the wide-issue superscalar models, thus providing better battery life. This is due to the fact that broadcasting over the issue queue to wakeup and select instructions consumes a considerable amount of power [7] in the superscalar model. Indeed, the hardware based dynamic scheduling of the superscalar is very effective in providing high degrees of instruction level parallelism, but such advanced technologies inevitably consume a large amount of power which shortens its battery life. This is why major embedded processor manufacturers such as ARM, MIPS, and the FPGA manufactures which also make soft cores such as Microblaze (from Xilinx) or Nios II (from Altera) hesitate to bring up active transition to the superscalar model in the embedded domain.

Despite of this power problem, future embedded applications require more complex operations eventually consuming extra power, which would in turn make the computing capability of the current conventional embedded processors not acceptable in the near future. With this forecast, it is crucial to provide more powerful embedded processors which could manage limited battery life efficiently for future applications. For that purpose, we propose a new processor model called Simultaneous Thin-Threading.

3 Simultaneous thin-threading: SMT for embedded processors

Simultaneous multithreading was originally developed to solve the long latency and pipeline stalling problem of the superscalar architecture. Applied on top of the existing superscalar architecture, the SMT design allows the
processor to fetch instructions from multiple threads. Therefore, after the fetching stage, all pipeline resources are shared between instructions from different threads [3, 4, 8]. This eventually supplies more independent instructions (Instruction-Level Parallelism) in a pipeline since instructions from the different threads (Thread-Level Parallelism) are independent to each other without having any data dependencies.

However, while the SMT approach solves the long latency problem and enhances utilization of superscalar processors, it still inherits the upper mentioned power consumption problem of the dynamic scheduling superscalar. Therefore, using the previously proposed SMT design in the embedded system is not suitable due to the large amount of power consumption.

In this paper, we propose the Simultaneous Thin-Threading (STT) technique, which is developed based on the SMT architecture and tuned to be suitable as embedded processors. The main idea is that our STT still supports multiple running programs and consequently supplies more independent operations to schedule, but with a very tiny dispatch queue. This expedites program execution from the multithreading with less hardware in order to reduce power consumption.

The design is mainly motivated by the fact that the most power demanding hardware structure in the traditional SMT is the dispatch queue. One critical problem of the scheduling logic could be sought in that it decides the scheduling of program using complex hardware structures, thus consuming a considerable amount of power. In this point, our design intends to reduce the size and complexity of the dispatch queue.

A conceptual design of the Simultaneous Thin-Thread processor is depicted in Fig. 1. Instead of having a single thread running as in the superscalar architecture, it supports simultaneous execution of multiple threads. This eventually provides more independent instructions inside a dispatch queue and can exploit higher degrees of instruction level parallelism. In our STT models for embedded systems, we target to reduce the power consumption by having a smaller dispatch queue and issue width. Our main contribution is that we can achieve equal or higher degree of ILP even with a small size of hardware by supporting multiple thread execution. The orig-
inality and creativity of the STT design lays in the fact that this is the first embedded processor with the multithreading feature in the research literatures. The main differences between STT and SMT can be found that STT is implemented with smaller issue width and dispatch queue in order to reduce the power consumption. Therefore, the STT design is more suitable and adequate as low-power embedded processors.

The notation \((x.y.z)\) in Fig. 1 and the following figures indicates: the first term \((x)\) specifies the architecture model and the second \((y)\) and the third \((z)\) numbers correspond to the issue width and dispatch queue size respectively.

4 Experimental results and performance analysis

In order to validate the proposed design, an accurate modeling process and simulation has been performed. The performance evaluation will be tested mainly on a simulator which basically measures the expected power consumption and program execution time. We have simulated two models of STT with Media benchmarks; \textit{STT-2} stands for a two-thread execution model whereas \textit{STT-4} indicates a four-thread execution model. Throughout the paper, the performance evaluation is performed on the ALPSS simulator \cite{6} which has been designed based on the SimpleScalar 3.0 \cite{1} and Wattch \cite{2}. We have chosen the MiBench \cite{5} suite as the benchmark programs for our simulation as MiBench is specially designed to target embedded applications \cite{5}.

In Fig. 2, we have compared six different architectures including the baseline superscalar processor \((SS.8.128)\) and two models for existing embedded processor architectures \((EP-SS \text{ and } EP-MT)\). EP-SS represents an embedded superscalar architecture which has a very limited parallel issue structure (4-way issue with a 16 entry dispatch queue). EP-MT models an even-based embedded multi-thread architecture which does not allow out-of-order issue within a thread.

We have measured the performance of each architectural model in terms of four different categories. The first metric is the IPC (instruction per cy-
cle), which shows processing capability of the processor. The second diagram shows normalized power consumption per cycle. The third diagram and forth diagram show the EPI (energy per instruction) and the EDP (energy delay product), respectively. The EDP is calculated by dividing the EPI by the IPC; this metric shows the power consumption considering the CPU computation capability. As we are aiming to design a processor with increased performance while maintaining minimum power consumption in the embedded domain, the EDP will be the suitable metric representing this aim.

As indicated in the results, the smallest configuration \((STT-2.4.8)\) shows the lowest power consumption per cycle. However, the configuration also results in the worst performance in terms of IPC. Looking at the EDP of \(STT-2.4.8\), the power consumption scaled to performance is higher than that of \(SS.8.128\). However, the other two STT models with dispatch queue sizes 16 and 32 show better results in the EDP than the baseline superscalar model. As indicated in the results, both of the current embedded processor models \((EP-SS\) and \(EP-MT)\) fail to provide high performance and energy efficiency.

Fig. 3 presents the performance results of four-thread execution model \((STT-4)\). As indicated in the graphs, the simulation showed similar results to that of the two-thread execution model. The smallest configuration with dispatch queue size of 8 \((STT-4.4.8)\) still shows the lowest EDP, while \(STT-4.4.16\) and \(STT-4.4.32\) achieves better results than \(SS.8.128\). Considering the results of EDP, \(STT-4.4.32\) shows the best performance, even more than that of \(STT-2.4.32\), and reduces the EDP values by 54.6%.

Fig. 3. Performance results with four-thread execution model

In summary, we can see in these results that the new STT technique, when having a dispatch queue size of 16 or 32, increases the IPC while having less power consumption than the superscalar model. Looking at the EDP metric, the STT extends performance almost twofold compared to the superscalar model. Considering the amount of performance improvement we need for future embedded applications, the STT method would be quite useful for improving the abilities of the embedded processor.
5 Conclusions

In this paper, we have proposed the Simultaneous Thin-Thread architecture which inherits the existing simultaneous multithreading techniques, but tuned for embedded processors. To the best of our knowledge, this is the first research paper to present the performance results of multithreaded embedded processors.

As mentioned earlier, future embedded applications will demand more computing performance, while need more power-efficient architecture. The STT technique targets these both requirements; the STT architecture will increase performance by exploiting ILP thorough multithreading, while consuming less power than the superscalar method.

The performance evaluation is based on the cycle-level, execution-driven performance simulator. We used 10 applications in 4 different categories, using the MiBench suite. The simulation results show that the proposed STT architecture shows better power consumption considering the CPU computation capability compared to the baseline, wide-issue superscalar model. It increases the performance significantly, almost twofold in maximum cases. We are confident to claim that the STT technique will be useful in embedded processor design.