Popular Science Summary

Memory Energy Optimizations for IoT

Ricardo Gómez

In battery operated devices, such as mobile phones, power consumption is an important constraint, as it has a direct impact on the battery lifetime. Trying to extend the battery lifetime as much as possible has been one of the major targets in device development for the past years. Together with advances in the battery technology, several techniques, usually referred as low-power design techniques, have been developed in order to reduce the power consumption of circuits.

In this Master Thesis, two optimizations that aim to reduce the power consumption has been proposed. This techniques has aimed the reduction of the power consumption caused by the memories.

The first proposal has been to reduce the size of the main memory. This technique is based on the fact that smaller memories consume less power. This desired size reduction can be achieved by compressing the contents in the memory. More concretely, the program code stored in the main memory unit has been compresed to reduce the overall memory size.

Well-known examples of data compression algorithms are WinZip, WinRar, 7zip, etc. In this Master Thesis, a more basic compression algorithm approach called Huffman compression has been implemented. Huffman compression is a compression algorithm which is based on the principle that assigning shorter binary words to the most frequent bytes and longer words to the less frequent bytes, a size reduction can be achieved. This approach is similar than the one used in Morse Code, where the most frequent letters (for example, the ‘e’) are substituted by the shortest codewords (in the case of ‘e’, a single dot).

In the implemented system, the memory contents are compressed offline, i.e. before the processor has been programmed. Then, a hardware decompressor is placed between the memory and the processor, so the compressed code is decompressed before is fetched to the processor.

The second proposal is based on powering off unused regions of the memory. By removing the supply power to these unused blocks, the standby power of them can be almost eliminated.

In most embedded processors, the main memory is physically implemented in a single block with a single power supply connection. This impedes powering off certain regions of the memory. For this reason, the first step has been partitioning the memory into smaller physical banks. These banks can be later powered on and off independently, achieving the desired behavior.

The power supplied to the memory banks is controlled by the system via the following protocol. When a new object (for example, an array of numbers) has to be allocated, the main program checks whether there is enough powered on memory or not. In case there is not enough powered memory, a request is generated and handled by a hardware block. This hardware block pauses the program execution and powers on the requested memory bank.

These techniques have been designed and implemented for JOP, a Java Optimized processor. The resulting system has been verified in a FPGA (Field Programmable Gate Array). Furthermore, an ASIC (Application Specific Integrated Circuit) has been designed using 65nm technology to evaluate the proposed optimizations.