*W*isconsin's *I*nterleaved *M*ultithreaded *P*rocessor

Bryan Berns
Jacob Petranak
Jordan Wenner
Parikshit Narkhede
Suman Mamidi

Project Overview

WIMP is a 16-bit interleaved multithreaded processor designed to exploit data parallelism inherent in many multi-media DSP applications. It handles four physical threads that are interleaved to share the hardware resources. The hardware threads are run in a round-robin scheme on a four-stage pipeline. The ISA contains a total of five classes of instructions, including: arithmetic and logical, load and store, jump, thread control, and system instructions. WIMP uses a subword parallel architecture with a limited form of predication applied to jump instructions. Threads can be initialized from another thread, with each thread capable of only killing itself. Synchronizing threads is used with a wait instruction that creates a barrier for threads to wait for each other. Thread 0 is used by the system, with threads 1-3 running applications. In addition, the system is capable of handling interrupts and exceptions with the aid of the system kernel that will be provided with the processor. The kernel is also responsible for ensuring that printed information from threads is stored in a buffer and printed out correctly. Lastly, the kernel is used for file transfers. Support for input and output devices will allow keyboard and SPART (Special Purpose Asynchronous Receiver Transmitter) interfacing. The keyboard is connected to the board to send and execute various commands to the processor. VGA is implemented on the system, and uses Video RAM Segment 1 and 2 in order to store all of the pixels that can be displayed on the monitor. An assembler was developed to create output formats acceptable for the board to read for developing applications. These file formats include: visual binary view for debugging, verilog, XES, and binary. The assembler is created with C++ and is capable of supporting many pseudo instructions. Macros were also developed for passing data to and from functions, printing strings and registers, and getting character inputs from WIMPTERM. WIMPTERM is a window that communicates with the board. It allows drag and drop support in order to easily download a binary output file from the assembler onto the board. WIMPTERM displays commands sent from the keyboard to the board. WIMPTERM also has image viewing capability that updates as a benchmark is processing. Benchmarks include parallelizable applications like image thresholding, bit plane slicing, image smoothing, and image filtering. Image thresholding involves comparing a pixel to a particular threshold value. If the pixel is greater than the threshold, the pixel is set to white; otherwise, it's set to black. Bit place slicing takes a pixel, then AND's the pixel with a bit slice value. The bit slice value is a byte in size, with only one bit set and the rest cleared. If the result of the AND operation is zero, the pixel is set to black; otherwise, the pixel is set to white. Image filtering uses a 3x3 pixel window and moves it through an image. Each pixel the image goes through is replaced with the weighted sum of the 3x3 window. Different values used in the 3x3 window results in applying different types of filters. In addition, time redundant fault tolerant programs are done. This program seeks to detect transient faults and re-execute the program if faults are found. The program run on WIMP will execute the program on two different physical threads and uses stores as checkpoints. The third thread will be used to monitor the checkpoints. If the result isn't consistent, then the application is restarted in the threads.