Summary
Multiscalar Processors

Mrinal Nath
January 31, 2005

Summary Part 1

What are the problems solved by this paper? (50 words)
It is normally difficult for the compiler to find much parallelism in a sequential program, since the compiler has to be conservative. This paper proposes TLS, so that compilers can create parallel threads (aggressively) even if all the dependencies are not known at compile time. The architectural support to enable TLS is based on the writeback invalidation based cache coherence protocol. The paper proposes extension of the basic protocol, so that threads can be executed in parallel and any dependency violations are detected.

What are the approaches attempted by this paper? (50 words)
This paper uses a combination of software (compiler) and hardware techniques to achieve the goals. The compiler produces epoch numbers for parallel tasks or epochs. The cache coherence hardware is enhanced (some extra states and coherence messages are added, and some bits are added to a cache line) so that any violations of the data dependencies are detected when the cache lines are updated. Accordingly, any threads which are executing speculatively with wrong data are squashed. The scheme proposed in the paper is scalable, and works well for systems of all sizes (SMT to shared memory multiprocessors).

What are the main conclusions of this paper? (50 words)
According to the results presented in the paper, significant speed-ups are possible using TLS. Also the overheads that are required to implement the scheme are small, and can be probably done in hardware.