

## Highest Performance Computing Machines

Y.N. Patt, University of Michigan (USA)

In recent years, many researchers have suggested that high performance execution of single instruction streams will result from dealing with complexity at the compiler level and making the hardware as simple as possible.

The opposite view has been taken by a few: drive all the complexity into the hardware, allowing the compiler to be much simpler. I suggest that optimum performance will require the best efforts of both the compiler and the hardware.

In this talk, I will introduce the HPS execution model which can and should provide substantial

support at the hardware level. I will report the result of some of our studies and suggest how HPS's effectiveness can be enhanced with appropriate, sophisticated compiler support.

The model involves dynamic scheduling (now called Superscalar) of multiple instructions to multiple multi-cycle functional units (now called Superpipelined). It involves out-of-order execution, deep pipelines without blocking, a good run-time branch predictor, and a fast checkpoint retirement mechanism.