High Performance Smith-Waterman Local Sequence Alignment

MAXELER Technologies

This is a summary of a presentation at the Symposium on Verification at the University of Tokyo, Japan, 2 February, 2011.

About the Authors

Akira Fukui is a graduate student in the Fujita Lab at the University of Tokyo.

Summary

We present an dataflow engine-based acceleration of the Smith-Waterman algorithm for local sequence alignment. It is used for similarity searching in bioinformatics protein sequence databases, although often less precise tools, such as BLAST, are preferred because of its high computational cost.

This presentation reports a Maxeler acceleration project which achieves 240,000 MCUPS (million cell updates per second) in a 1U server, representing a 128x speedup over the industry-standard SSE-accelerated SSEARCH implementation (8-core Intel X5570 2.93GHz Nehalem EP), using a Maxeler MaxNode with 4 MAX2 DFEs.

As illustrated in *Figure 1*, Smith-Waterman has a wavefront dependency structure, in which parallelism is present along the diagonals of the matrix. To implement this efficiently, our array of cell update processing units (PU in *Figure 2*) operates on a stream of independent tasks in order to exploit deep pipeline parallelism despite the tight dependence structure of the algorithm, and the dynamically-varying number of parallel operations along the diagonals of the matrix (see *Figure 3*). This loop-tiling optimization, sometimes called c-slowing, is analogous to simultaneous multithreading (SMT) in microprocessors and enables full utilisation of the PU pipelines, which deliver one cell update per cycle. Additional optimizations reduce the word length of the arithmetic operations to match the maximum range of values.

This design was undertaken by a visiting graduate researcher over a 3 month period, with no prior knowledge of the Smith Waterman algorithm, dataflow engine design or Maxeler tools.

Further performance improvements can be achieved by more aggressive optimization of the bit-widths of the data types in the design.

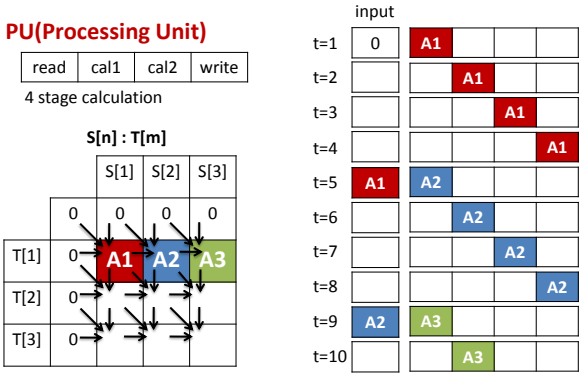

Figure 2: The dependence structure limits available parallelism.

	-	S[1]	···	S[i]	···	S[n]
-	V(0,0)=0	V(1,0)=0 E(1,0)=0	V(•,0)=0 E(•,0)=0	V(i,0)=0 E(i,0)=0	V(•,0)=0 E(•,0)=0	V(n,0)=0 E(n,0)=0
T[1]	V(0,1)=0 F(0,1)=0	V,E,F	···	···	···	···
···	V(0,•)=0 F(0,•)=0	···	···	···	···	···
T[j]	V(0,j)=0 F(0,j)=0	···	···	V(i,j)	···	···
···	V(0,•)=0 F(0,•)=0	···	···	···	···	···
T[m]	V(0,m)=0 F(0,m)=0	···	···	···	···	V,E,F

Figure 1: Comparison of two sequences S and T, requires a cell update that depends on left, upper and diagonal neighbours.

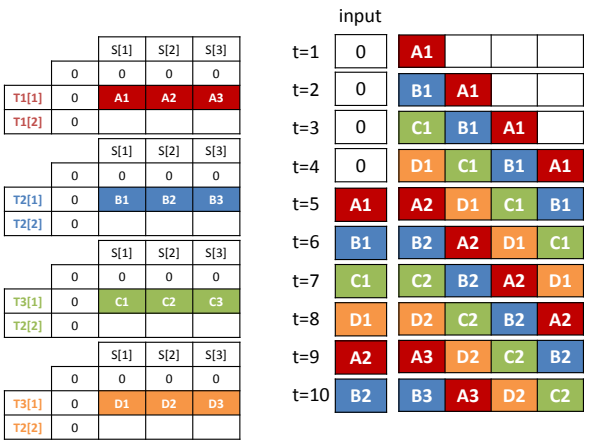

Figure 3: Interleaved scheduling of matching tasks on the processor array.