HAT: A HARDWARE ASSISTED TOP-DOC INVERTED INDEX COMPONENT

S. Kagan Agun
Information Retrieval Laboratory
Department of Computer Science
Illinois Institute of Technology
agunsal@iit.edu

Ophir Frieder
Information Retrieval Laboratory
Department of Computer Science
Illinois Institute of Technology
ophir@ir.iit.edu

ABSTRACT
A novel Hardware Assisted Top-Doc (HAT) component is disclosed. HAT is an optimized content indexing device based on a modified inverted index structure. HAT accommodates patterns of different lengths and supports a varied posting list versus term count feature sustaining high reusability and efficiency. The developed component can be used either as an internal slave component or as an external co-processor and is efficient in resource demands as the component controllers take only a minimal percentage of the target device space leaving the majority of the space to term and posting entries. A Very High Speed Integrated Circuit (VHSIC) Hardware Description Language (VHDL) is used to model the HAT system.

Categories and Subject Descriptors
H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing – *indexing methods*. B.7.1 [**Integrated Circuits**]: Types and Design Styles – *algorithms implemented in hardware*. B.6.3 [**Logic Design**]: Design Aids – *hardware description languages*.

General Terms
Hardware assisted inverted index file design.

Keywords
Inverted index file, hardware support.

1. INTRODUCTION
Invert index files are used to support the efficient searching of documents. Static index pruning [3] often reduces the number of posting entries stored in the index while still providing comparable accuracy in query processing. By storing the posting entries of only those documents for which a given term appears frequently in, the posting list size of the index is potentially dramatically reduced, improving runtime performance. By implementing such an approach in hardware, the described HAT system provides either an internal slave component or an external co-processor that aids in high speed document searching.

Unlike prior hardware component support for document searching [1, 2, 4], HAT focuses on a chip that maintains a pruned inverted index rather than on filtering based on pattern matching. Mapping the highly-accessed software structure onto a chip reduces the processing time associated with index access and simplifies the maintenance.

SIGIR'03, July 28–August 1, 2003, Toronto, Canada
ACM 1-58813-646-3/03/0007.

2. THE HAT ARCHITECTURE
The HAT external interface (Figure 1) includes an address bus (*Address*) and a bi-directional data bus (*Data*), both can be instantiated to any width. The output lines *RW*, *Enable*, and *Ready* are used to handle the memory access. HAT also includes asynchronous *Reset* and *Halt* signals where reset initializes the HAT and halt terminates the operation of the HAT. The *clock* signal is the system clock. The *Error* output indicates an unrecoverable error state. The term *Found*, *Lstatus*, and *Rstatus* signals are used to stack up to 16 HAT components in an array.

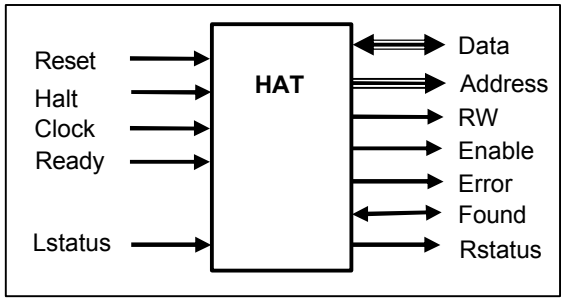

Figure 1. The HAT External Interface

The HAT system architecture (Figure 2) consists of two main components, a term matching processor and an array of posting processors, and a master controller including a device interface. Each posting unit handles one posting list set which stores, in a sorted order, the top n documents. These units are connected to

the internal data bus and controller signals, are highly parallel, and support master – slave operations. While a query is retrieving data from a posting list unit, another unit can execute sorted list update operations. The HAT Controller is the master controller that distributes the work and manages the communication with the main processor. The high performance bus interface can be a PCI bus or any other bus available on the market for reconfigurable computing.

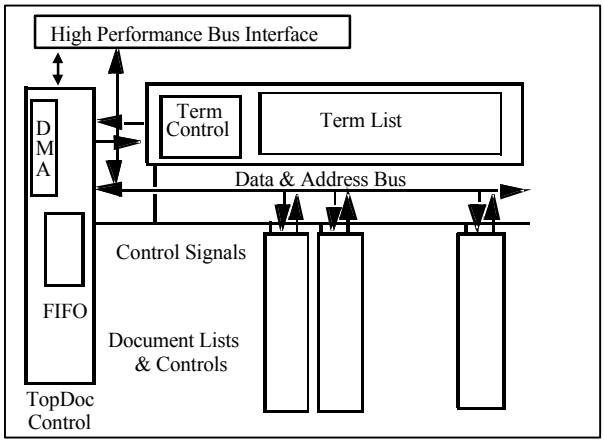

Figure 2. The HAT System Architecture

Pattern matching involves a character-by-character comparison with terms having all of their characters compared against the search pattern simultaneously. HAT exploits parallelism in term-matching to reduce the document search time. In Figure 3, we illustrate the parallel term-matching units.

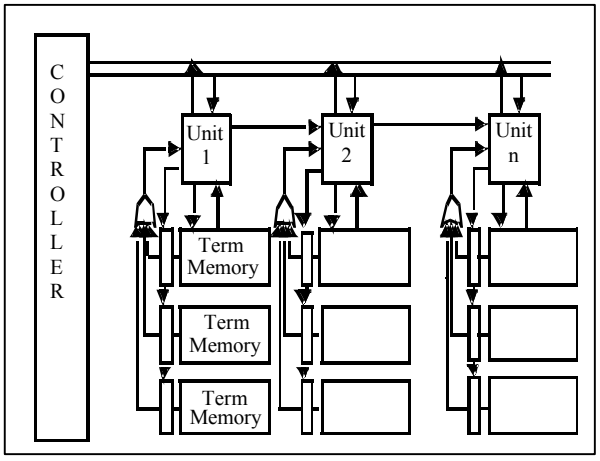

Figure 3. Term-matching Units

The posting list and term matching structures are similar. In addition to the scanning of documents, the posting list unit also inserts a document into the proper location to maintain a sorted order. Using 8 bits for the weight provides a count from 0 to 255, leaving 24 bits for the document identifier.

3. RESULTS

We used the Altera Leonardo Spectrum, a suite of high-level design tools for hardware synthesis, to design HAT. In Table 1, we illustrate the logic cell usage and speed requirement of HAT components for the target device Cyclone EP1C20T400C, the smallest device in this category. It is worth noting that the control logic of the components requires only a small amount of cell resources. For example, the TermUnit and PostUnit controllers use 124 and 178 logic cells that are less than 1% of the target device where the memory bits are the main resource. Using internal chip memory as a cache also achieves high performance for HAT components. These components exploit parallelism for a given application through reconfigurable and flexible hardware units.

Table 1. Resources claimed by HOT components.*

Component	LogicCell (LC)	Memory (bits)	Frequency (MHz)
Generic DualPort Memory (256x32)	--	8192 (3.13%)	356.2
TermFIFO + Gen Mem(256x32)	124 (0.62%)	8192 (3.13%)	146.1
TermUnit + TermFIFO(256x32)	178 (0.89%)	8192 (3.13%)	90.6
PostUnit + GenMem(256x32)	93 (0.46%)	8192 (3.13%)	110.8
HAT (100 terms)	8340 (41.58%)	212992 (81.25%)	79.7

(*Cyclone EP1C20T400C (20,060 LC, 294,912 Memory bits))

4. CONCLUSION

As with all software mapped onto hardware components, using a hardware implementation of the Top-Doc algorithm reduces query-processing times. HAT was developed using a reconfigurable and reusable hardware architecture design approach and is intended for use in consumer commodity personal computers to support document search applications. Currently, commercialization discussions are ongoing.

5. REFERENCES

[1] Fast Search Chip, www.fastsearch.com, 2002.

[2] Mak, V.W., K. C. Lee, and O. Frieder, "Exploiting Parallelism in Pattern Matching: An Information Retrieval Application," ACM Trans. on Information Systems, 9(1), Jan. 1991, pp. 52-74.

[3] Soffer, A., et al., "Static Index Pruning for Information Retrieval Systems", ACM SIGIR, Sept. 2001, pp. 43-50.

[4] TextFinder FDF 4, Paracel Corporation, www.paracel.com/products/textfinder.html, 2002.