

Characteristics of the Xerox 1100 Machines upon which the Gabriel Benchmarks Were Performed

All three members of the Xerox 1100 family are custom microcoded processors. The Interlisp-D virtual machine is built around a compact 8-bit "bytecode" instruction set, the opcodes of which are implemented by a combination of microcode and macrocode. Not all bytecodes are supported directly in each member by microcode; the alternative is a trap out to a standard Lisp function. Above the level of the instruction set, all three members of the family appear identical to the Interlisp-D programmer. The implementation is such that a memory image can be compatibly run on any of the machines, without any change.

An Interlisp pointer is an address in a 24-bit virtual address space; a "quantum map" indexed by the high bits of the address provides information for type decoding. Additionally, litatoms (symbols) and immediate numbers (integers in the range of -2^{16} to $2^{16}-1$) live in a reserved portion of the address space; integers of larger magnitude (within the range -2^{31} to $2^{31}-1$) are "boxed"; floating-point numbers, which are in IEEE 32-bit format, are also boxed. All three machines have a 16-bit memory bus and 16-bit ALU; however, the bytecodes tend to hide the actual word size from the programmer. The virtual address space is broken down into units of 512-byte pages, and the three machines have different degrees of hardware assist for virtual memory management and instruction fetch.

Cons cells are cdr coded in a manner described in D. Bobrow and D. Clark, "Compact Encodings of List Structure", ACM Trans. on Prog. lang. and Systems, Vol 1 No 2, p266 October 1979. A cell of 32 bits is used to store a cons -- typically 24 bits for the car, and 8 bits for an encoding of the cdr. The encoding covers the four cases where (1) the cdr is NIL, or (2) the cdr is directly on the same page as the cons cell, or (3) the cdr is contained in another cell on the same page as the cons cell, or (4) the cons cell is itself a full indirect pointer, which can address an ordinary two-cell slot on any page (the space normally used for the car is used to address a 64-bit cell elsewhere; this is to allow for RPLACDs when there is no more free cells on the same page as the cell being updated). All cons cells are cdr-coded, independent of how they are created, and as a consequence the "average size" of such a cell is considerably less than 64 bits.

Strings and arrays are implemented as a fixed-length header, with one field pointing to a variable-length memory chunk taken from an area which is separately managed. To run some of the benchmarks, we used Interlisp's Common Lisp array utility package. Additionally, Interlisp permits the user to define new first-class fixed-length data types, with corresponding entries in the quantum map mentioned above; for example, a STREAM is implemented as a record structure with 19 pointer fields and assorted integer fields of 16 bits or less.

Garbage collection is patterned after Deutsch and Bobrow, "An Efficient, Incremental, Automatic Garbage Collector" CACM, July 1976. A reference count is maintained for every collectible pointer (in addition to immediate pointers, litatoms are not reclaimed in Interlisp-D). Updates to non-stack cells in data structures (i.e., the CAR slot of a CONS cell, or the value-cell of a global variable) require updates to the reference count. The reference counts are maintained separate from the objects in a hash table, which is generally very sparse; and the updating is normally done within the microcode that effects the update operations. Reclamations are

performed frequently, and involve scanning the stack area and augmenting the reference counts by a "stackp" bit; then scanning the reference count table reclaiming any entry which has a count of 0 and no reference from the stack (and possibly additional pointers whose reference count goes to zero as a result of such a reclamation); and finally re-scanning the table to clear the "stackp" bits. The scan through the reference count table looking for 0-count entries corresponds roughly to the scan of the marked-bits table in a Mark-and-Sweep collector; however, the scan of the stack is infinitesimal in time compared to a full "mark" phase, and thus a reclamation typically runs in well under a second.

The internal architecture of the stack is a variant of the "spaghetti stack" model described in Bobrow and Wegbreit "A Model and Stack Implementation of Multiple Environments, Comm. ACM, Vol. 16, No. 10, Oct. 1973, pp. 591-603. The stack area is currently limited to 128KB.

The particular configurations upon which the benchmarks were run are as follows:

Xerox 1100: (Dolphin). 4K words of 40-bit microstore; microinstruction time 180ns; hardware assist for macro-instruction fetch; hardware memory map for up to 8MB of virtual space; hardware stack (for stack tip); memory access is 1-to-4 words (64 bits) in about 2us. The particular unit used in the benchmarking runs had 1.8MB of real memory attached, but 2MB has been in standard delivery.

Xerox 1108: (DandeLion) 4K words of 48-bit microstore; microinstruction time 137ns; hardware assist for macro-instruction fetch; hardware assist for virtual memory management (memory map is kept in non-paged real memory); memory access is 1 non-mapped 16-bit word in 411ns, but a random 32-bit cell access in about 1.2us. The stack is held in real, non-mapped memory. The particular unit used in the benchmarking runs had 1.5MB of real memory attached.

Xerox 1132: (Dorado) 4K words of 34-bit high-speed ECL microstore; microinstruction time 64ns; hardware instruction fetch unit; hardware memory map for up to 32MB of virtual space; 4Kilowords of high-speed ECL memory cache permit memory access of one 16-bit word in 64ns, and a cache-reload of 256 bits takes about 1.8us (additional details on the cache and memory organization may be found in D. Clark, B. Lampson, and K. Pier: "The Memory System of a High-Performance Personal Computer", IEEE Transactions on Computers, vol C-30, no. 10, Oct 1981). The particular unit used in the benchmarking runs had 2MB of real memory attached.

Note that the benchmarks were not run on the 1108-111 (DandeTiger), which has considerably more memory and control store than the basic 1108, and which also has a floating-point processor.