THE LSI-11/23 CONTROL STORE MICROARCHITECTURE

Burt Hashizume and William N. Johnson
Small Systems Engineering
DIGITAL EQUIPMENT CORPORATION
Maynard, Massachusetts

ABSTRACT

The LSI-11/23, an LSI version of Digital Equipment's PDP-11/34, is the first microcomputer with the performance and features of a midrange minicomputer. Three forty-pin packages containing five NMOS LSI chips form the basis for a complete CPU on an 5" x 8" board with extensive memory management features and both single and double precision floating point instructions (see Figure 1). This paper describes one of the features, the control store and control sequencer*, that enabled the LSI-11/23 to obtain two and a half times the performance of the LSI/11, while maintaining a flexible microarchitecture for instruction set extensions.

Figure 1. LSI-11/23 CPU

*Digital Equipment Corporation has a patent application pending before the U.S. Patent Office relating to the structure and operational features of the control store.

THE LSI-11/23

Unlike many other new 16-bit microcomputers, the 11/23 does not present a new architecture to the user. One of the design goals was complete compatibility with other PDP-11s and with the PDP-11/34 in particular. Thus, the 11/23 complements the PDP-11 family as an 11/34 performance machine at roughly LSI-11 cost (see Figure 2). In addition, not only is the 11/23 architecturally upward compatible with the LSI-11, but it also utilizes the same electrical bus structure so it can use the same backplanes, memory modules, and I/O modules. One can literally remove the CPU module from an 11/03 system, replace it with an LSI-11/23, and run exactly the same software on the same hardware. This implies that the vast amount of existing PDP-11 systems and application software can run without any changes on the LSI-11/23.

Figure 2. PDP-11 Family

The core of the LSI-11/23 consists of three custom NMOS LSI chips: a data path chip, a memory management chip, and a control store chip (control chip). The data path chip contains the 16-bit user register file, 16-bit scratch registers, and a 16-bit arithmetic logic unit. The memory management chip contains the memory management registers, all of the relocation logic, and the floating point register file. The control chip contains both a mask-programmable control store and a microsequencer. Multiple control chips with
different microcode patterns can be concatenated in a system to provide instruction set extensions, operating system assists, etc. Control is transferred between control chips via microjump instructions. The 11/23 LSI chips communicate with each other and with surrounding logic via two buses: a 16-bit microinstruction bus (MIB) and a 16-bit data/address bus (DAL). The MIB bus transfers 16-bit microinstructions and control information, while the DAL bus communicates not only address and data, but interrupt and abort information as well (see Figure 3).

Control Store

The control store structure typical of many minicomputer designs consists of an instruction register, a mapping ROM or PLA, read/write or read-only memory for the microprogram control store, and a microsequencer (see Figure 4). The macro instruction is initially loaded into the instruction register. This feeds into a mapping PLA or ROM which decodes the instruction and generates the starting address of the instruction's execution routine in the control store. Sometimes there may be several stages of mapping PLA or ROM for decoding addressing modes in addition to the opcodes.

Sequencing through the microstore can be implemented by either of two basic techniques. The first technique utilizes a microprogram counter to address the microinstructions with explicit microjumps to alter the microflow. This method is generally used in low-end vertically microcoded machines such as the LSI-11. The second technique uses a next address field in the microinstruction with a 9-bit next address field.

In the LSI-11/23 each control chip is a self-contained control unit with its own control store and sequencing logic (see Figure 5). The design of the chip is modular, however, so that additional control store can be added just by adding more control chips to the system. The heart of the control chip is an integrated PLA/ROM array consisting of 552 microwords. The microword exhibits characteristics of both vertical and horizontal microcodes by combining a highly encoded 16-bit microinstruction with a 9-bit next address field.
Twenty-five percent of the 11/23 microstore is comprised of PLA terms, and the remaining seventy-five percent are ROM words. The PLA has 25 inputs: 9 microaddress inputs and 16 data bits from a PLA Input Register (PIR). The PIR can be loaded with the macroinstruction or any type of microsequencing information. The ROM has only the 9 microaddress inputs. The address space is partitioned such that the PLA terms only respond to addresses 0 to 127 and the ROM words only to addresses 128 to 511.

Each PLA term and each ROM word contain both a microinstruction and next address field. This implies two unique characteristics: 1) the output of the PLA is an executable microinstruction, not a microaddress into the microstore, and 2) PLA terms and ROM words may be mixed in any order because the next address field allows jumping between the two address spaces with no time penalty. Thus, the 11/23 control chip has collapsed the traditional two-tier control architecture of separate ROM and PLA arrays into a single integrated structure. This microprogram can use PLA terms to execute the addressing and initial emulation phases of the macrocycle, and the ROM bank to execute the inline microcode for longer routines such as multiply and divide.

This integrated PLA/ROM structure has another unique advantage over traditional control store architectures. Multiple terms of the PLA can be activated simultaneously and their outputs 'AND-ed' to provide powerful bit-override controls at the microinstruction level, rather than just at the microaddress level.

**APPLICATIONS**

Because of the nature of the PDP-11 instruction opcode distribution, where the opcode field is of varying length from four bits to sixteen bits, the PLA's are ideally suited for the instruction decode. Parts of the AND array of the PLA are programmed to decode instruction classes, addressing modes, and individual instructions which enable certain OR terms that contain the desired microword to be executed. The integration of the PLA into the microstore eliminates the need to wait for dispatching, and hence provides for immediate execution of the first microinstruction in the appropriate addressing or emulation sequence. Most of the PDP-11 arithmetic and logic instructions are emulated by a similar microinstruction, and accordingly can be executed in a single microcycle following instruction fetch with no decoding overhead.

In order to reduce the number of PLA terms required to perform an exhaustive decode of the instructions and addressing modes, the "override" technique of multiple PLA terms is used. In many cases where two classes of operations are similar but not identical, it is possible to fully implement only the first class of operations with discrete PLA terms, and then program just one more PLA term which only decodes the difference between the two classes. This additional term can be activated by the second class of operations to alter the microinstruction normally programmed for the first class by overriding one or more bits of the microinstruction output. An example is the class of byte versus the class of word instructions. Both the macroinstruction and the microinstruction formats are defined such that the byte and word variations differ by the same bit. One PLA term per microinstruction type is programmed so it is enabled for both the word and byte variations of a given macroinstruction, and one additional PLA term is programmed so it is enabled for all word-type macroinstructions in general.

Consider the case of the move word (MOV) and move byte (MOVB) macroinstructions (see Figure 6). When MOV is being executed, only the MOV/MOVB PLA term will be enabled, resulting in a uMOV microinstruction. When a MOV macroinstruction is being executed, both the MOV/MOVB PLA and the generic word-type PLA will be enabled. The former results again in a uMOV microinstruction and the latter generates a bit pattern that overrides the byte/word microinstruction bit of the uMOV microinstruction to create a uMOV microinstruction. Note that the one word-type PLA term is usable for all the byte/word instructions in the entire macroinstruction set. Thus, the 34 word/byte PDP-11 instructions can be decoded using only (17 + 1) PLA terms, rather than 34 terms, resulting in a great savings of microstore space.

<table>
<thead>
<tr>
<th>MACROINSTRUCTION</th>
<th>PLA OUTPUT</th>
<th>MICROINSTRUCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOV R1→R0</td>
<td>0010000010000000</td>
<td>( \mu \text{MOV R1→R0} )</td>
</tr>
<tr>
<td>MOV R1→R0</td>
<td>0010000010000000</td>
<td>( \mu \text{MOV R1→R0} )</td>
</tr>
<tr>
<td>MOVB R1→R0</td>
<td>0001000000000000</td>
<td>( \mu \text{MOVB R1→R0} )</td>
</tr>
<tr>
<td>CMP R1→R0</td>
<td>0100000000000000</td>
<td>( \mu \text{CMP R1→R0} )</td>
</tr>
<tr>
<td>CMP R1→R0</td>
<td>0100000000000000</td>
<td>( \mu \text{CMP R1→R0} )</td>
</tr>
<tr>
<td>CMP R1→R0</td>
<td>0100000000000000</td>
<td>( \mu \text{CMP R1→R0} )</td>
</tr>
<tr>
<td>CMP R1→R0</td>
<td>0100000000000000</td>
<td>( \mu \text{CMP R1→R0} )</td>
</tr>
<tr>
<td>CMP R1→R0</td>
<td>0100000000000000</td>
<td>( \mu \text{CMP R1→R0} )</td>
</tr>
</tbody>
</table>

Figure 6. PLA Override Examples

System interrupts, such as device interrupt requests, line clock interrupts, bus errors, and memory management aborts, will occur asynchronously to instruction execution. A number of these must be tested at the end of every instruction execution. Many microarchitectures test for interrupts with a conditional microjump that penalizes the end of every macroinstruction cycle with one microcycle. The 11/23, however, uses the PLA struc-
ture to test for interrupts with no time penalty. While the last microinstruction of a macroinstruction routine is being executed, the PLA Input Register is loaded with the interrupt information. If interrupts are pending, the PLA will decode for the appropriate service routine in the next microcycle and generate the first service microinstruction. If none are pending, the resultant PLA translation will instead generate the first microinstruction in the macroinstruction routine. Thus testing for interrupts is invisible in terms of execution time.

This control structure can also be utilized in various ways to implement small lookup tables. One method is to load the PLA Input Register with the input data to be translated, and program the PLA terms to decode the combinations that are desired. This is excellent for fairly sparse tables because of the large number of data inputs allowed in the PIR (up to 16 bits), and because only one PLA term per table entry needs to be utilized. An override PLA term or a default PLA term can be used to trap all illegal input combinations and initiate a common routine for fault handling.

Although the 11/23 control chip has no dedicated microsubroutine hardware, the capability exists using the PLA Input Register. On a microsubroutine "call", the return address is loaded into an internal scratch register in the data chip with a literal microinstruction, and the microsubroutine starting address is taken from the next address field. When returning from the microsubroutine, the return address in the scratch register is transferred to the PIR on the control chip and used to translate back to the original microprogram.

These examples reflect some of the ways this powerful control structure has been used in the current LSI-11/23 product. Many other possibilities exist, which will be used as other extended instruction sets are added to the LSI-11/23.