

creates three instructions with a cost of 41, and another that encodes the load and store into a memory operand of a single instruction and costs just 17. The cheapest pattern is selected.

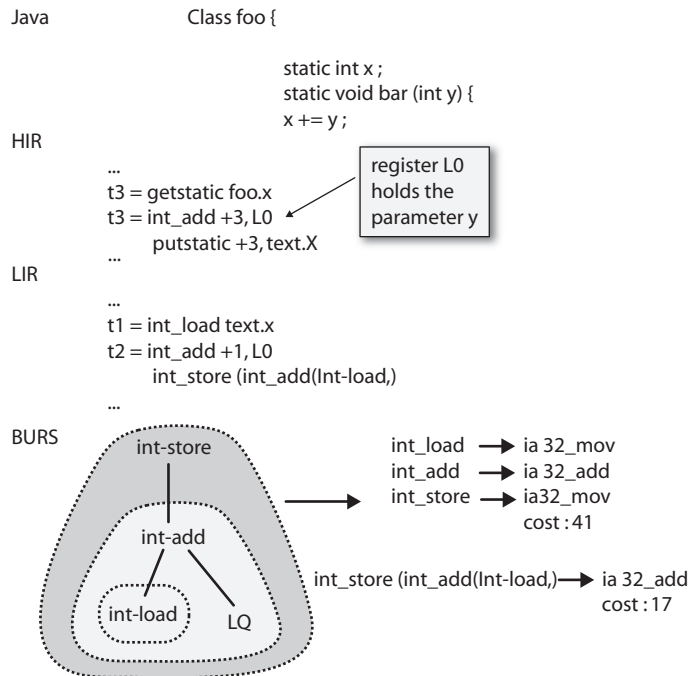


FIGURE 10-6. BURS instruction selection

After instruction selection, the unlimited number of registers need to be mapped onto the finite number of registers provided by a real machine. This process is known as register allocation. When there are not enough registers, values can be held in memory on the stack; swapping an actual register for a value on the stack is known as spilling and filling. A register allocator needs to minimize spills and fills, as well as taking into account any architectural requirements—for example, ones that multiply and divide must occur using certain registers.

Jikes RVM has a linear scan register allocator that performs quick register allocation, but with the possibility of generating extra copy and memory operations. This is a greater problem when the number of registers available on the machine is small. With increases in the number of registers and improvements to the linear scan algorithm, for certain code it is less clear that it would be beneficial to perform a more expensive register allocation.

The final part of MIR instruction creation is instruction scheduling. Scheduling separates instructions in order to allow the processor to exploit instruction-level parallelism.

At the level of MIR, few optimizations are performed. Because of the large number of side effects associated with machine instructions, it is difficult to reason and therefore optimize the