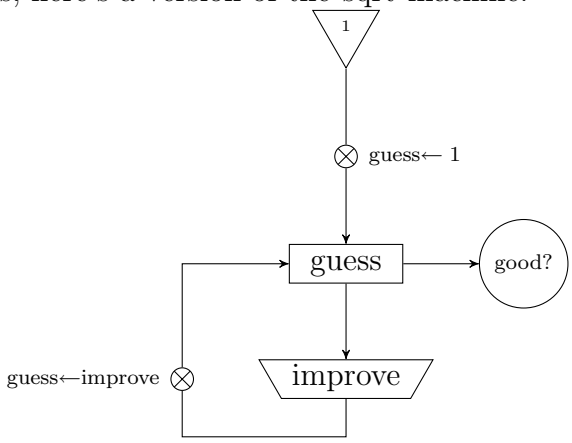
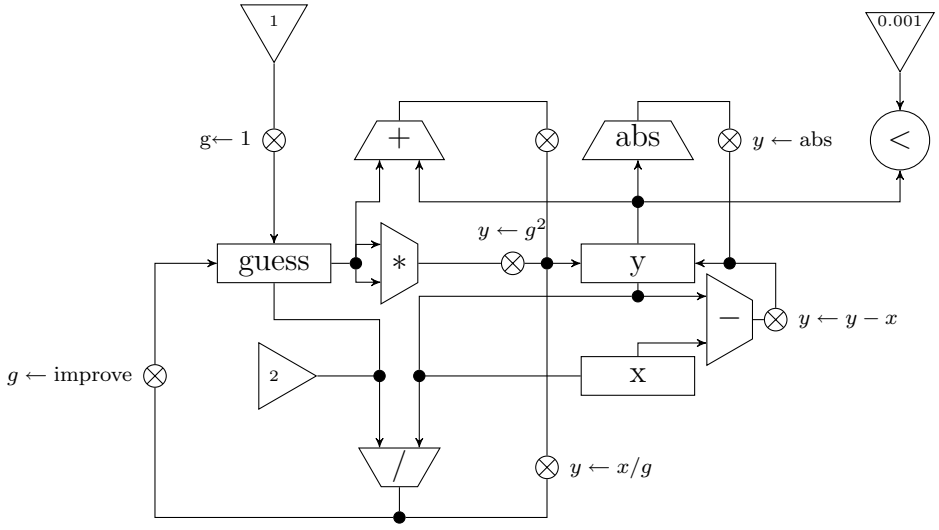
By assuming that good? and improve operations are available as primitives, here's a version of the sqrt machine.




```
(controller
   (assign guess (const 1))
 test-guess
   (test (op good?) (reg guess))
   (branch (label sqrt-done))
   (assign guess (op improve) (reg guess))
   (goto (label test-guess))
 sqrt-done)
```

Here's the expanded version in terms of arithmetic operations.




```
(controller
   (assign guess (const 1))
 test-guess
```

```
  (assign y (op *) (reg guess) (reg guess))
  (assign y (op -) (reg y) (reg x))
  (assign y (op abs) (reg y))
  (test (op <) (reg y) (const 0.001))
  (branch (label sqrt-done))
  (assign y (op /) (reg x) (reg guess))
  (assign y (op +) (reg y) (reg guess))
  (assign guess (op /) (reg y) (const 2))
  (goto (label test-guess))
sqrt-done)
```