```ocaml
(* $Id: muldivrem-trace.ml,v 1.1 2012-02-07 19:44:53-08 - - $ *)

let double number = number + number

let rec mul' (multiplier, powerof2, multiplicand') =
    if powerof2 > multiplier
    then multiplier, 0
    else let remainder, product =
                  mul' (multiplier, double powerof2, double multiplicand')
             in  if remainder < powerof2
                 then remainder, product
                 else remainder - powerof2, product + multiplicand'

let mul (multiplier, multiplicand) =
    let _, product = mul' (multiplier, 1, multiplicand)
    in  product

let rec divrem' (dividend, powerof2, divisor') =
    if divisor' > dividend
    then 0, dividend
    else let quotient, remainder =
                  divrem' (dividend, double powerof2, double divisor')
             in  if remainder < divisor'
                 then quotient, remainder
                 else quotient + powerof2, remainder - divisor'

let divrem (dividend, divisor') = divrem' (dividend, 1, divisor')

let div (dividend, divisor) =
    let quotient, _ = divrem (dividend, divisor)
    in quotient

let rem (dividend, divisor) =
    let _, remainder = divrem (dividend, divisor)
    in remainder

;;
#trace mul     ;;
#trace mul'    ;;
#trace div     ;;
#trace rem     ;;
#trace divrem  ;;
#trace divrem' ;;

```

```
bash-1$ ocaml
        OCaml version 4.02.1

# #use "muldivrem-trace.ml";;
val double : int -> int = <fun>
val mul' : int * int * int -> int * int = <fun>
val mul : int * int -> int = <fun>
val divrem' : int * int * int -> int * int = <fun>
val divrem : int * int -> int * int = <fun>
val div : int * int -> int = <fun>
val rem : int * int -> int = <fun>
mul is now traced.
mul' is now traced.
div is now traced.
rem is now traced.
divrem is now traced.
divrem' is now traced.
# mul (745, 1033);;
mul <-- (745, 1033)
mul' <-- (745, 1, 1033)
mul' <-- (745, 2, 2066)
mul' <-- (745, 4, 4132)
mul' <-- (745, 8, 8264)
mul' <-- (745, 16, 16528)
mul' <-- (745, 32, 33056)
mul' <-- (745, 64, 66112)
mul' <-- (745, 128, 132224)
mul' <-- (745, 256, 264448)
mul' <-- (745, 512, 528896)
mul' <-- (745, 1024, 1057792)
mul' --> (745, 0)
mul' --> (233, 528896)
mul' --> (233, 528896)
mul' --> (105, 661120)
mul' --> (41, 727232)
mul' --> (9, 760288)
mul' --> (9, 760288)
mul' --> (1, 768552)
mul' --> (1, 768552)
mul' --> (1, 768552)
mul' --> (0, 769585)
mul --> 769585
- : int = 769585
#
```

```
bash-1$ ocaml
        OCaml version 4.02.1

# #use "muldivrem-trace.ml";;
val double : int -> int = <fun>
val mul' : int * int * int -> int * int = <fun>
val mul : int * int -> int = <fun>
val divrem' : int * int * int -> int * int = <fun>
val divrem : int * int -> int * int = <fun>
val div : int * int -> int = <fun>
val rem : int * int -> int = <fun>
mul is now traced.
mul' is now traced.
div is now traced.
rem is now traced.
divrem is now traced.
divrem' is now traced.
# div (876543, 123);;
div <-- (876543, 123)
divrem <-- (876543, 123)
divrem' <-- (876543, 1, 123)
divrem' <-- (876543, 2, 246)
divrem' <-- (876543, 4, 492)
divrem' <-- (876543, 8, 984)
divrem' <-- (876543, 16, 1968)
divrem' <-- (876543, 32, 3936)
divrem' <-- (876543, 64, 7872)
divrem' <-- (876543, 128, 15744)
divrem' <-- (876543, 256, 31488)
divrem' <-- (876543, 512, 62976)
divrem' <-- (876543, 1024, 125952)
divrem' <-- (876543, 2048, 251904)
divrem' <-- (876543, 4096, 503808)
divrem' <-- (876543, 8192, 1007616)
divrem' --> (0, 876543)
divrem' --> (4096, 372735)
divrem' --> (6144, 120831)
divrem' --> (6144, 120831)
divrem' --> (6656, 57855)
divrem' --> (6912, 26367)
divrem' --> (7040, 10623)
divrem' --> (7104, 2751)
divrem' --> (7104, 2751)
divrem' --> (7120, 783)
divrem' --> (7120, 783)
divrem' --> (7124, 291)
divrem' --> (7126, 45)
divrem' --> (7126, 45)
divrem --> (7126, 45)
div --> 7126
- : int = 7126
#
```

```
bash-1$ ocaml
        OCaml version 4.02.1

# #use "muldivrem-trace.ml";;
val double : int -> int = <fun>
val mul' : int * int * int -> int * int = <fun>
val mul : int * int -> int = <fun>
val divrem' : int * int * int -> int * int = <fun>
val divrem : int * int -> int * int = <fun>
val div : int * int -> int = <fun>
val rem : int * int -> int = <fun>
mul is now traced.
mul' is now traced.
div is now traced.
rem is now traced.
divrem is now traced.
divrem' is now traced.
# rem (876543, 123);;
rem <-- (876543, 123)
divrem <-- (876543, 123)
divrem' <-- (876543, 1, 123)
divrem' <-- (876543, 2, 246)
divrem' <-- (876543, 4, 492)
divrem' <-- (876543, 8, 984)
divrem' <-- (876543, 16, 1968)
divrem' <-- (876543, 32, 3936)
divrem' <-- (876543, 64, 7872)
divrem' <-- (876543, 128, 15744)
divrem' <-- (876543, 256, 31488)
divrem' <-- (876543, 512, 62976)
divrem' <-- (876543, 1024, 125952)
divrem' <-- (876543, 2048, 251904)
divrem' <-- (876543, 4096, 503808)
divrem' <-- (876543, 8192, 1007616)
divrem' --> (0, 876543)
divrem' --> (4096, 372735)
divrem' --> (6144, 120831)
divrem' --> (6144, 120831)
divrem' --> (6656, 57855)
divrem' --> (6912, 26367)
divrem' --> (7040, 10623)
divrem' --> (7104, 2751)
divrem' --> (7104, 2751)
divrem' --> (7120, 783)
divrem' --> (7120, 783)
divrem' --> (7124, 291)
divrem' --> (7126, 45)
divrem' --> (7126, 45)
divrem --> (7126, 45)
rem --> 45
- : int = 45
#
```