load'mult.ijs'
alu_64
3 : 0
NB. $a$ and $b$ are each 64 -bit summands
NB. the result is a 64-bit sum
' $\mathrm{a} \mathrm{b}^{\prime}=$. y .
(64\#2) rep (base $x:$ a) + base $x: b$
)
a
1111111111111111111111111111111111111110
base a
4.29497e9
signed_value a
|value error: signed_value
| signed_value a
load'arith.ijs'
signed_value a
_2
$a=:(64 \# 2)$ rep _2
a

```

``` b
00000000000000000000000000000010
alu_64 a;b
0000000000000000000000000000000000000000000000000000000000000000 c =: (64\#2) rep _1
c
```



``` alu_64 b; c
00000000000000000000000000000000000000000000000000000000000000001 alu_64 34
0000000000000000000000000000000000000000000000000000000000000111 not_equal
~:
last
_1\&from
a
1111111111111111111111111111111111111111111111111111111111111111110 1 from a
1
0 from a
1
_1 from a
0
mult1
3 : 0
('multiplicand' ; 'multiplier') =. y.
product=. 64\#0
count=. 32
while. 0 not_equal count
do. control =. last multiplier
if. control
```

            do. product =. alu_64 product ; multiplicand
            end.
        multiplicand =. (1 drop multiplicand),0
        multiplier =. 0 , _1 drop multiplier
        count =. <: count
    end.
    product
)
((62\#0), 1 1)

```

```

    (30#0),1 0
    O O O O O O O O O O O O O O O O O O O O O O O O O O O O O O 1 0
mult1 ((62\#0), 1 1); (30\#0),1 0

```

```

    signed_value mult1 ((62#0), 1 1); (30#0),1 0
    6
display
1!:2\&2
traced_mult1 ((62\#0), 1 1); (30\#0),1 0

```

```

+-------+-------------------------------------------------------------------------------------------------------------------------------------

```


```

+-----------+----------------------------------------------------------------------------------------------------------------------------------------------
|multiplier|O 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 11
+----------+-------------------------------------------------------------------

```



```

+------------+--------------------------------------------------------------------------------------------------------------------------------------------
|multiplier| 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 |
+----------+--------------------------------------------------------------------
|multiplicandlO 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 01
+------------+------------------------------------------------------------------------------------------------------------------------

```

```

+----------+---------------------------------------------------------------------

```

```

+------------+------------------------------------------------------------------------------------------------------------------
|multiplier|0000000000000000000000000000000001
+----------+-----------------------------------------------------------------------------

```













do. product \(=:(64 \# 2)\) rep (base \(x:\) product) + base \(x:\) b
end.
)
signed_value mult1m ((62\#0), 1 1); (30\#0),1 0
6
alu_32
3 : 0
NB. a and b are each 32-bit summands
NB. the result is a 32-bit sum
('a' ; 'b') =. y.
(32\#2) rep (base x: a) + base x: b
\(a=:(31 \# 1), 0\)
\(b=:(30 \# 0), 10\)
alu_32 a;b
0000000000000000000000000000
c =: 32\#1
alu_32 b;c
```



``` a
\(\begin{array}{lllllllllllllllllllllllllllllllll}1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0\end{array}\)
b
0000000000000000000000000000000000000010 C
\(\begin{array}{lllllllllllllllllllllllllllllllll}1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\end{array}\)
base alu_32 100 ; 28
128
mult2 ((30\#0), 11\()\); (30\#0), 10
```



``` mult2 (0,31\#1);0,31\#1
```



``` base mult2 \((0,31 \# 1) ; 0,31 \# 1\)
4611686014132420609
base 0,31\#1x
2147483647
(base 0,31\#1x)^2
4611686014132420609
base mult2m ((30\#0), 1 1); (30\#0),1 0```

