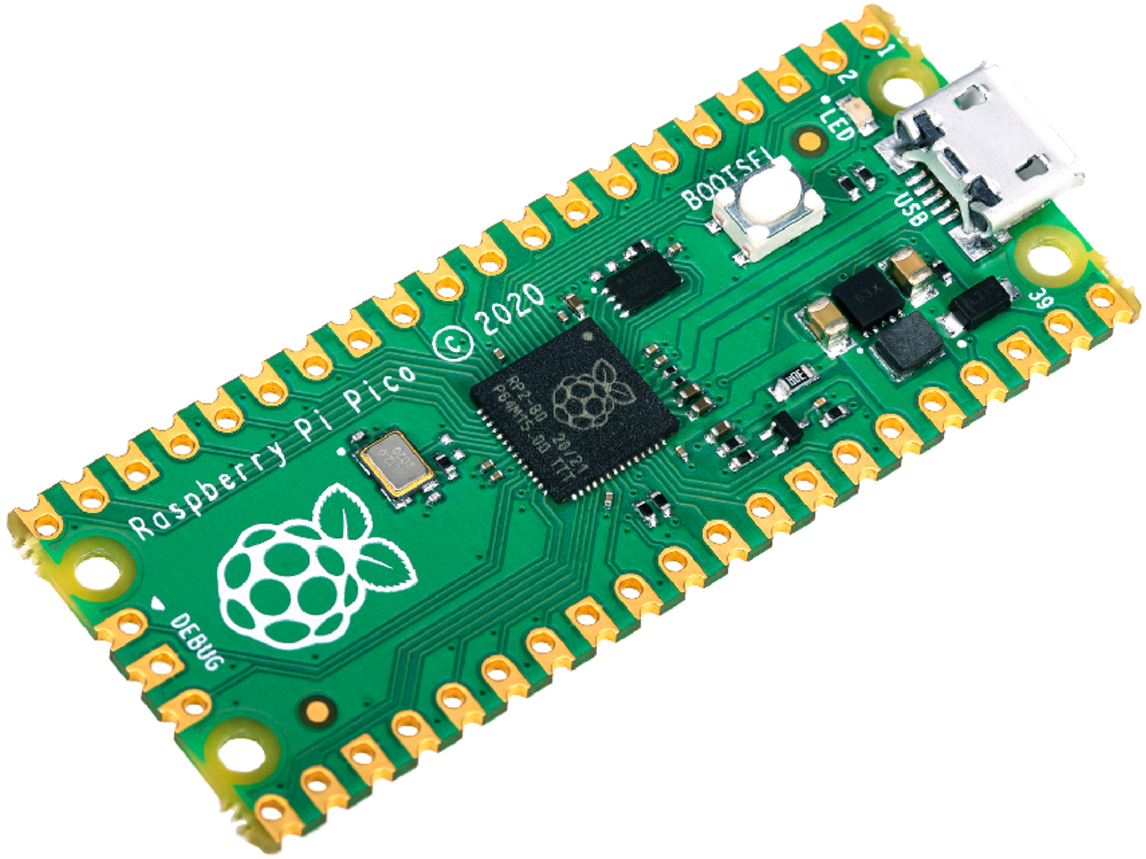
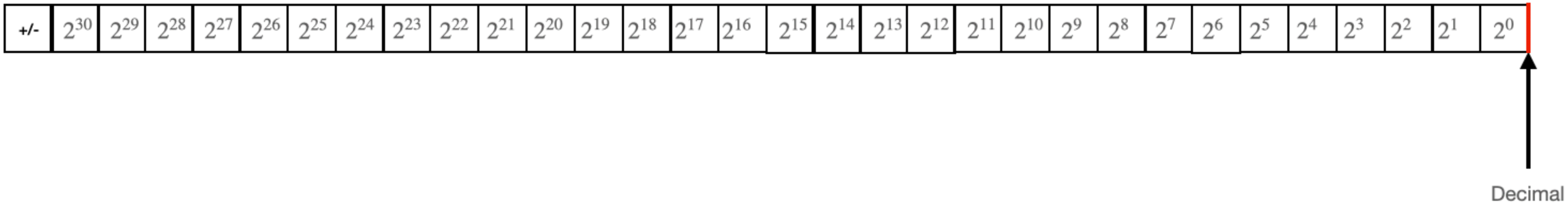


Raspberry Pi Pico (RP2040)

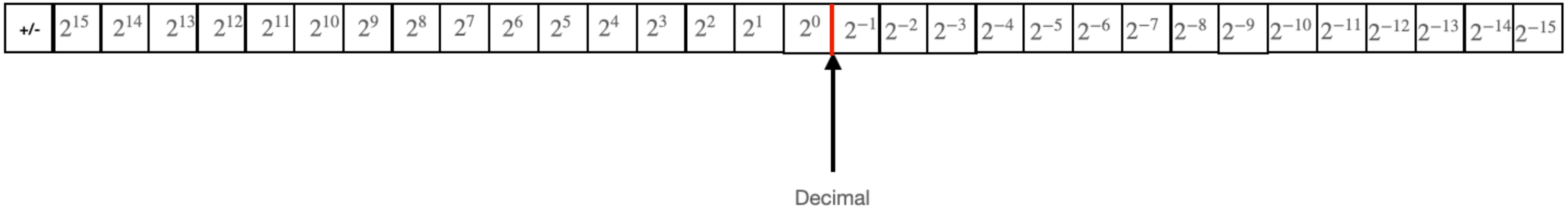
Lecture 6: Protothreads and Fixed Point

Recalling `signed int`



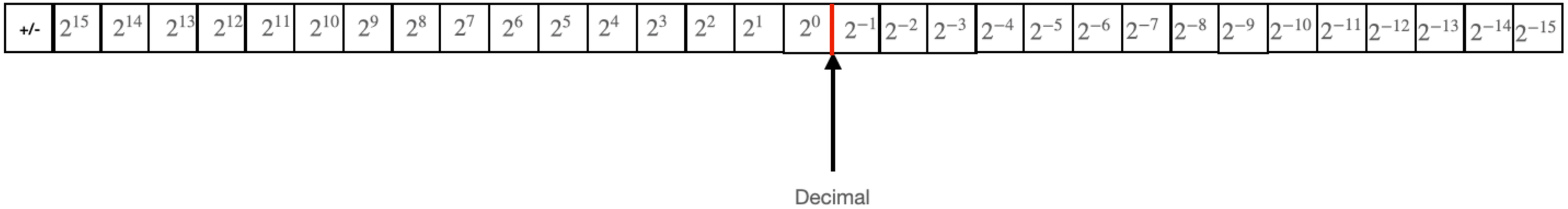
- Range of $\left[-2^{31}, \quad 2^{31}-1\right]$
- Resolution of 1

Introducing `fixed point`



- We imagine placing the decimal *somewhere else* in the signed int, precisely where is up to you, and depends on your range/resolution requirements
- For the particular example above, range of $\left[-2^{15},\quad 2^{15}-1\right]$
- For the particular example above, resolution of 2^{-15}
- We've traded **range** for **resolution**

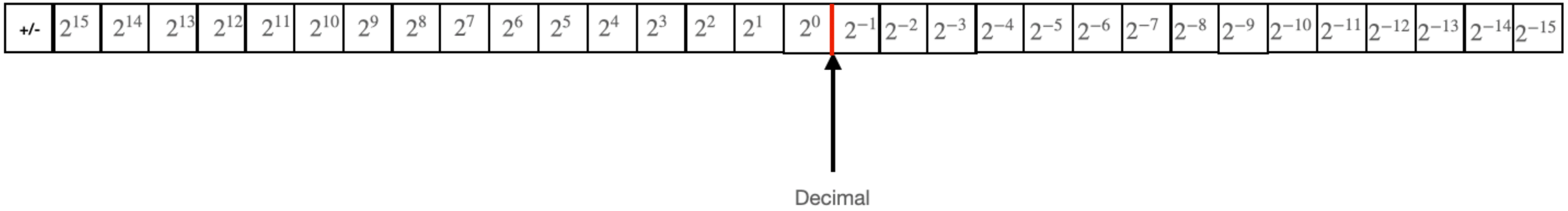
Creating a `fixed point` data type




```
typedef signed int fix15 ;
```

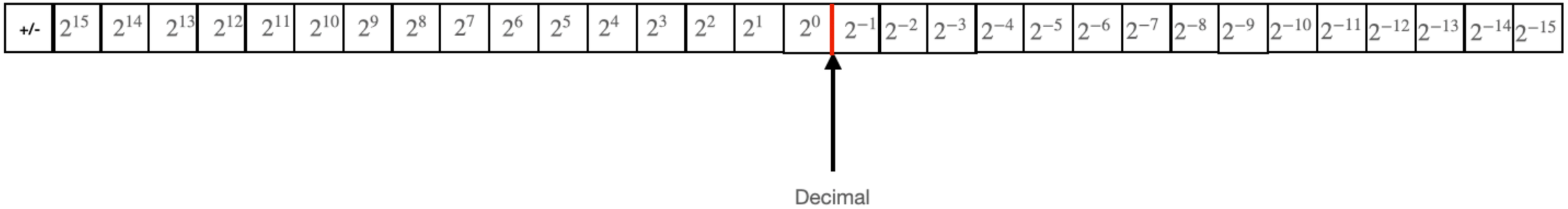
To the CPU, a fixed point data type *looks* like a signed int.

Arithmetic with `fixed point`



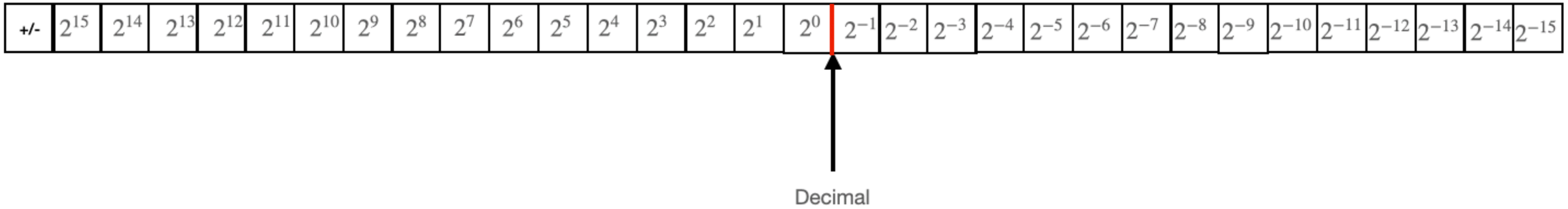
- For which arithmetic operations does this only affect our *interpretation* of the number? And for which arithmetic operations must we implement fixed-point specific macros?

Arithmetic with `fixed point`



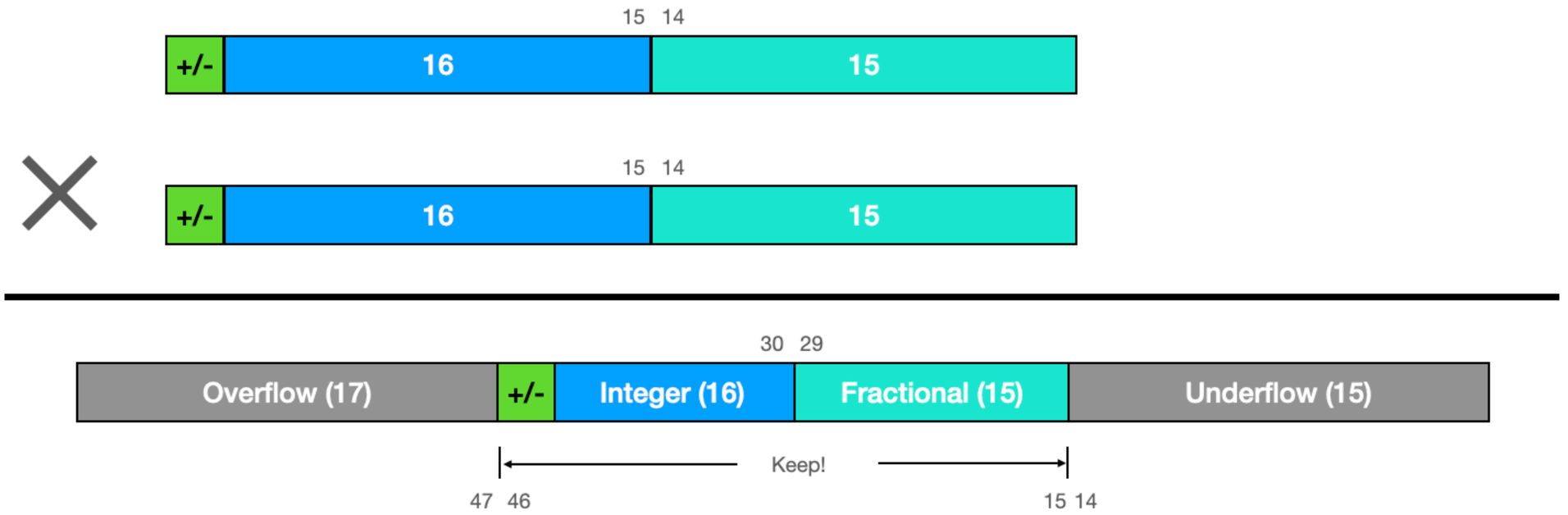
Do addition/subtraction work?

Arithmetic with `fixed point`

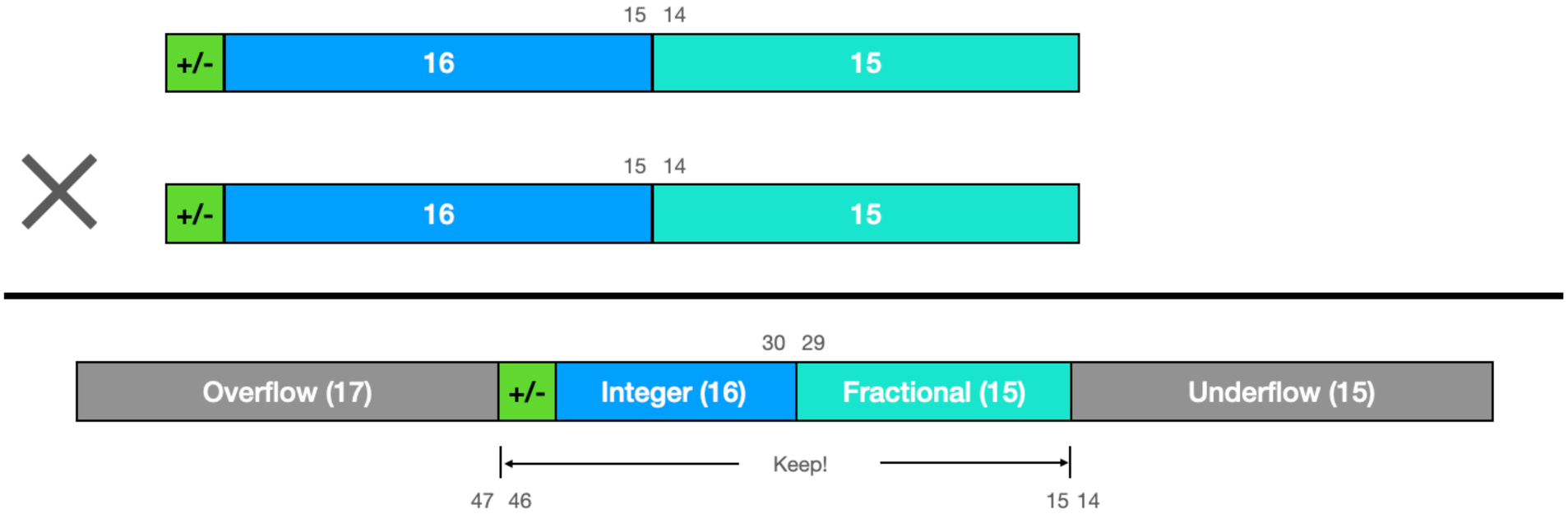


Does multiplication work?

Arithmetic with `fixed point`

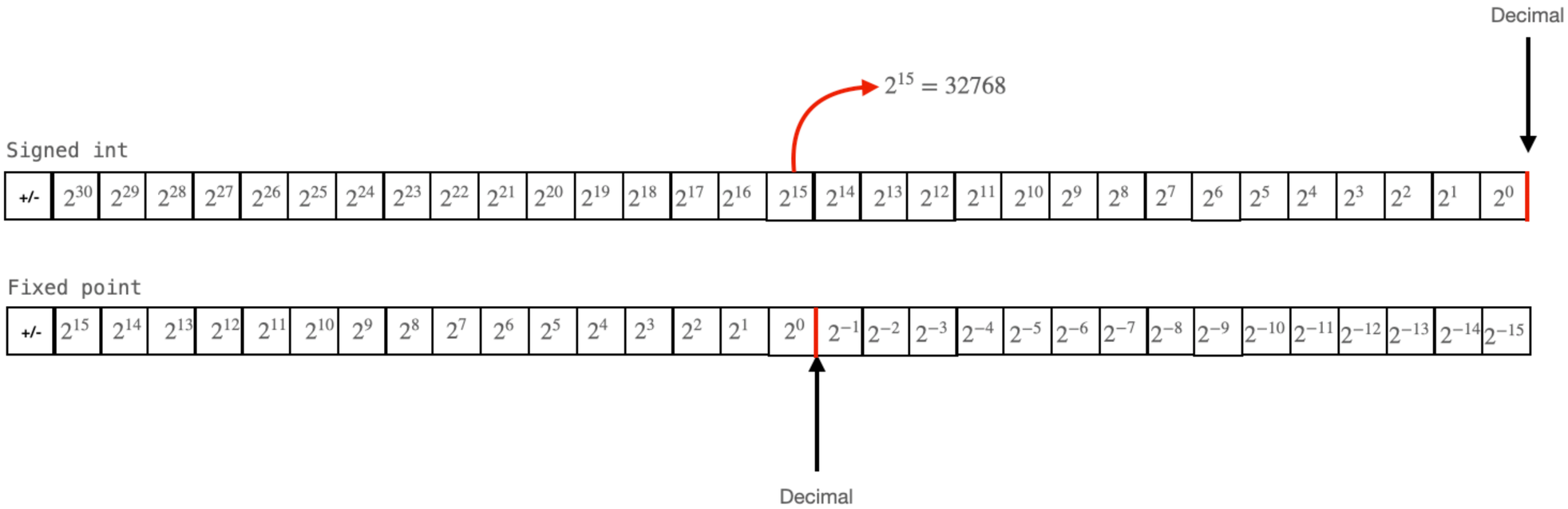


Arithmetic with `fixed point`




```
#define multfix15(a,b) ((fix15)(((signed long long)(a)*(signed long long)(b))>>15))

#define divfix(a,b) (fix15)( (((signed long long)(a)) << 15) / (b))
```

Type conversions to/from `fixed point`

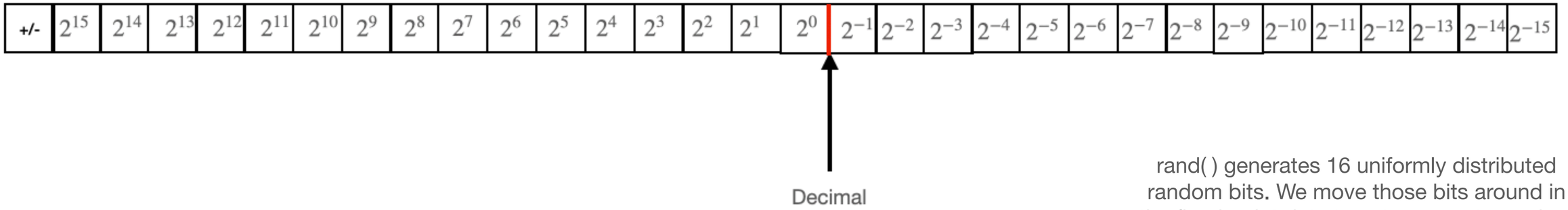



```
#define int2fix15(a) ((fix15)(a << 15))

#define fix2int15(a) ((int)(a >> 15))

#define float2fix15(a) ((fix15)((a)*32768.0))

#define fix2float15(a) ((float)(a)/32768.0)
```

Square root with `fixed point`

```
#define sqrtfix(a) (float2fix15(sqrt(fix2float(a))))
```

Random number generation with `fixed point`



rand( ) generates 16 uniformly distributed random bits. We move those bits around in the fixed point data type to generate random numbers in the range that we want. From top to bottom, the expressions on the left generate random numbers in the range [0, 1], [-1, 1], [-2, 2], [-1.5, 1.5]

```
rand_fix = (fix)(rand() & 0xffff) >> 16 ;
```

```
rand_fix = ((fix)(rand() & 0xffff) >> 15) - int2fix15(1) ;
```

```
rand_fix = ((fix)(rand() & 0xffff) >> 14) - int2fix15(2) ;
```

```
rand_fix = ((fix)(rand() & 0xffff) >> 15) + ((fix)(rand() & 0xffff) >> 16) - float2fix15(1.5)  ;
```

Speed comparison