Training Digital Signal Processing

ELETDS02

Fixed point calculations

brojz@hr.nl
muiko@hr.nl

Last week

- The IDTFT gives us an infinite number of coefficients of our **FIR filter**.
- To implement the filter in practice we need to apply **windowing**.
- Rectangular windowing might introduce **unwanted effects** in the frequency domain.
- **Different window formulas** exist that try to keep certain unwanted effects to a minimum. (Experiment with these!)

- **IIR filters** contain feedback (or are recursive).
- With only a few coefficients good results can be achieved .
- Might be unstable.

FIXED POINT CALCULATIONS

Fixed point versus floating point

- Floating point numbers:
  - Standardized in IEEE Standard for Floating-Point Arithmetic (IEEE 754)
  - Supported in almost any programming language
  - Big dynamic range, e.g. 32-bit single precision (`float` in C): 1.2E-38 to 3.4E+38
  - Precision varies with value, e.g. `float` has 23 significant bits, the precision depends on the position of the binary point which varies
  - Calculations take more time / chip area / power than integer calculations
- Fixed point numbers:
  - Not standardized and unsupported in almost any programming language
  - Small dynamic range (watch out for overflows)
  - Fixed precision
  - Calculations are almost the same as integer calculations

FIXED POINT EXAMPLES 1

Integer numbers are almost always represented in two's complement binary

$$x = 0100100000011000_b = 18456_d$$

Fixed point numbers are often represented in Q*m*.*n* format
Q*m*.*n* = a two's complement number with *m* bits before and *n* bits behind the binary point and an implicit sign bit

$$\boldsymbol{Q0.15}: x = 0.100100000011000_b$$
$$x = 2^{-1} + 2^{-4} + 2^{-11} + 2^{-12} = 0.56323_d$$

$$\boldsymbol{Q2.13}: x = 010.0100000011000_b$$
$$x = 2^{1} + 2^{-2} + 2^{-9} + 2^{-10} = 2.25293_d$$

$$\boldsymbol{Q5.10}: x = 010010.0000011000_b$$
$$x = 2^{4} + 2^{1} + 2^{-6} + 2^{-7} = 18.02344_d$$

FIXED POINT EXAMPLES 2

2's complement integer:

$y = 1100111111111000_b = (-)0011000000001000_b = -12296_d$

$\boldsymbol{Q0.15}$: $y = (-)0.011000000001000_b$

$y = -(2^{-2} + 2^{-3} + 2^{-12}) = -0.37524414_d$

$\boldsymbol{Q2.13}$: $y = (-)001.100000001000_b$

$y = -(2^{0} + 2^{-1} + 2^{-10}) = -1.500976563_d$

$\boldsymbol{Q5.10}$: $y = (-)001100.0000001000_b$

$x = -(2^{3} + 2^{2} + 2^{-7}) = -12.0078125_d$

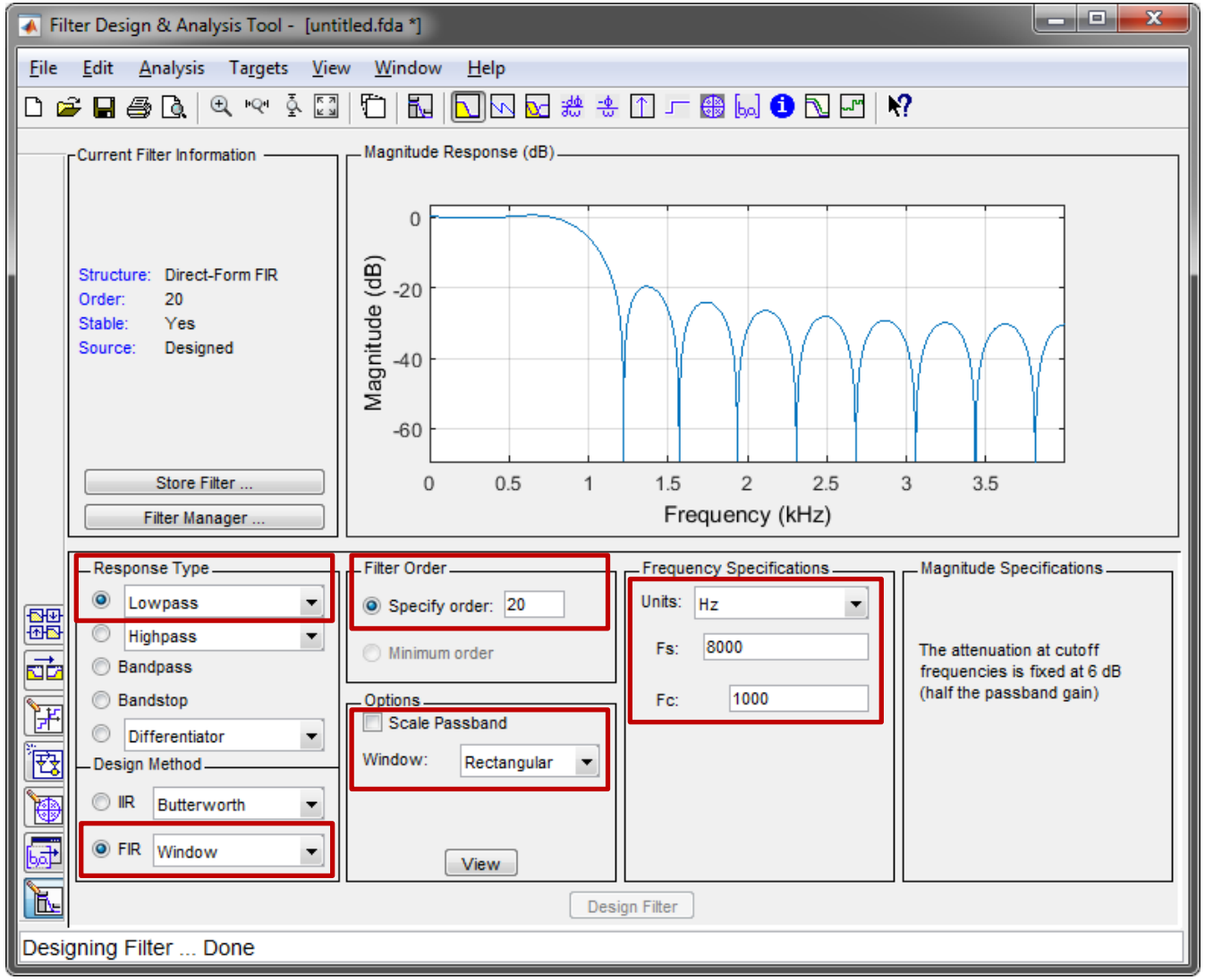
FIXED POINT MATLAB NOTATION

General notation	Matlab
$Q0.15$	$s16,15$
$Q2.13$	$s16,13$
$Q5.10$	$s16,10$

16 bits total of which 10 bits right of the point

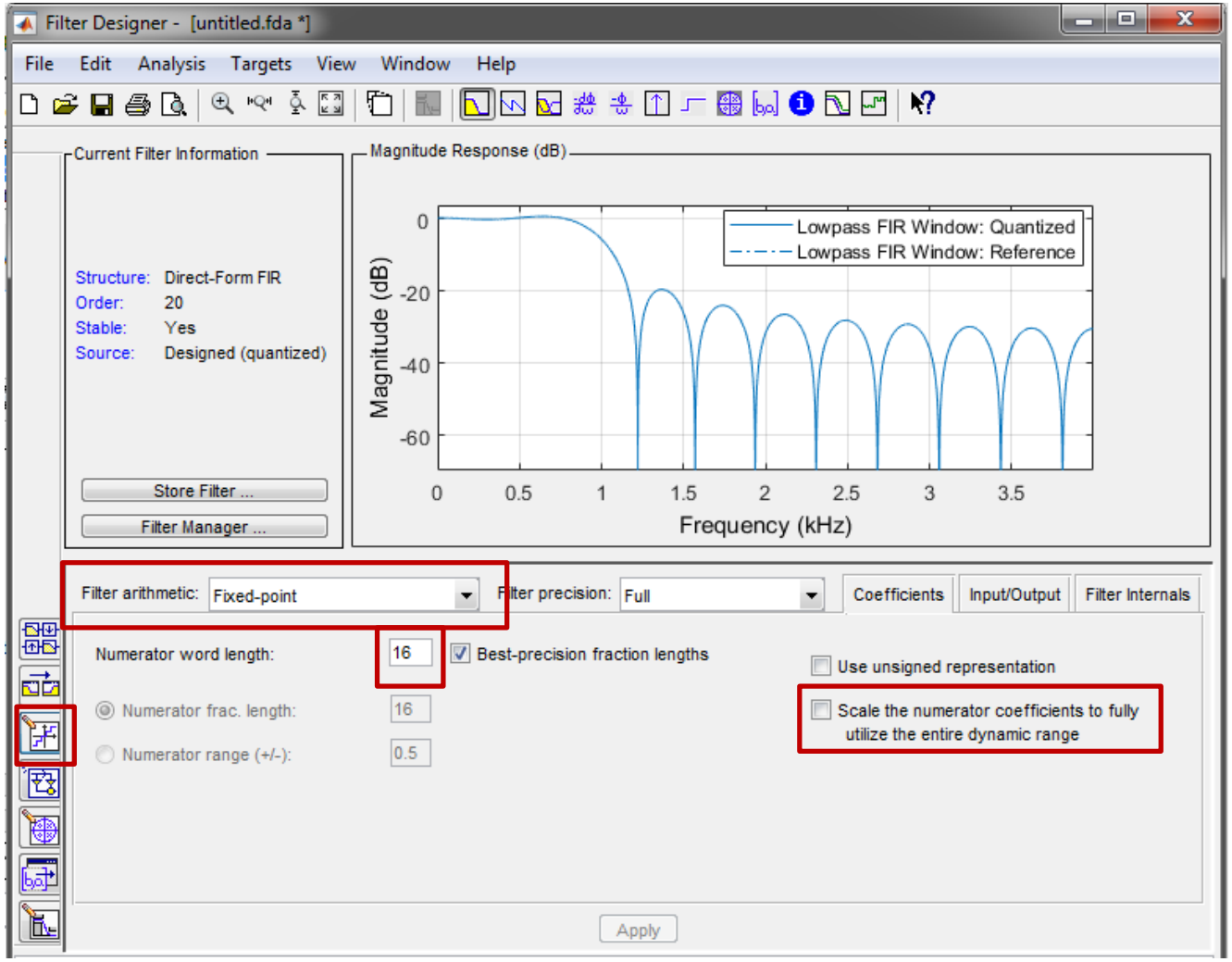
So 6 bits left of the point (including signbit)

FIR FILTER COËFFICIENTS 1



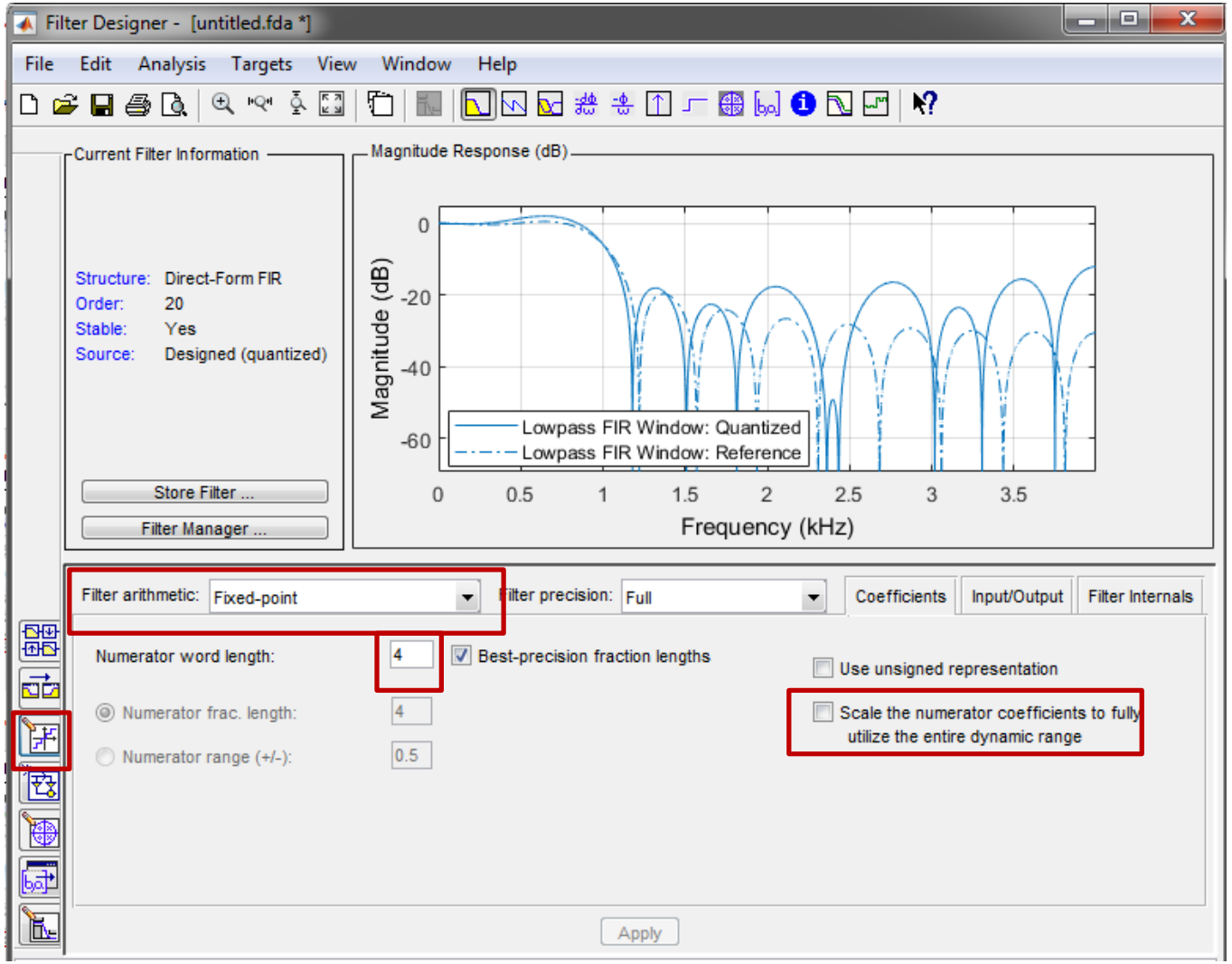
HOGESCHOOL ROTTERDAM

FIR FILTER COËFFICIENTS 2



HOGESCHOOL ROTTERDAM

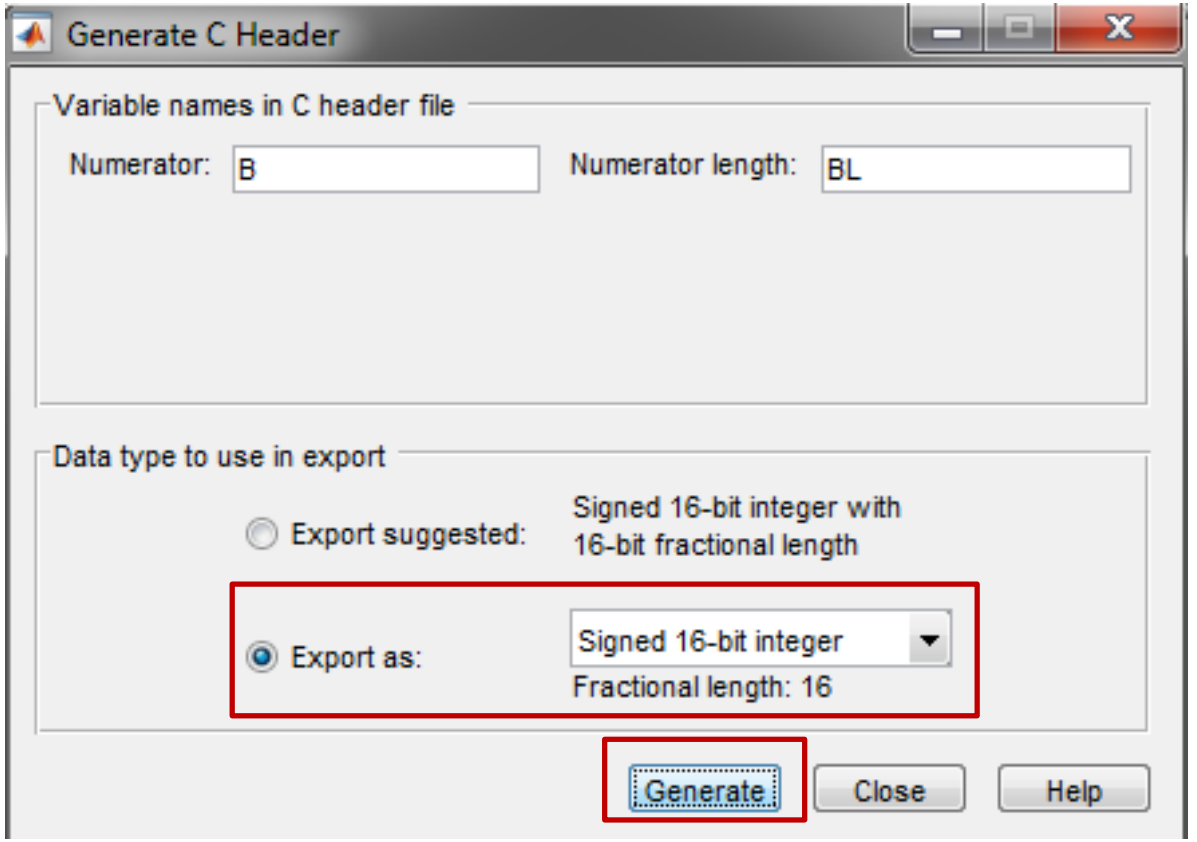
FIR FILTER COËFFICIENTS 3



HOGESCHOOL ROTTERDAM

FIR FILTER COËFFICIENTS 4

Targets -> Generate C Header



HOGESCHOOL ROTTERDAM

C-HEADER

```
]/*
  * Filter Coefficients (C Source) generated by the Filter Design and Analysis Tool
  * Generated by MATLAB(R) 9.0 and the DSP System Toolbox 9.2.
  * Generated on: 15-Sep-2016 15:23:37
  */

]/*
  * Discrete-Time FIR Filter (real)
  * -------------------------------
  * Filter Structure  : Direct-Form FIR
  * Filter Length     : 21
  * Stable            : Yes
  * Linear Phase      : Yes (Type 1)
  * Arithmetic        : fixed
  * Numerator         : s16,16 -> [-5.000000e-01 5.000000e-01)
  * Input             : s16,15 -> [-1 1)
  * Filter Internals  : Full Precision
  *   Output          : s33,31 -> [-2 2)  (auto determined)
  *   Product         : s31,31 -> [-5.000000e-01 5.000000e-01)  (auto determined)
  *   Accumulator     : s33,31 -> [-2 2)  (auto determined)
  *   Round Mode      : No rounding
  *   Overflow Mode   : No overflow
  */

/* General type conversion for MATLAB generated C-code  */
#include "tmwtypes.h"
]/*
  * Expected path to tmwtypes.h
  * C:\Program Files\MATLAB_2016a\extern\include\tmwtypes.h
  */
const int BL = 21;
]const int16_T B[21] = {
     2086,   1639,      0,  -2107,  -3477,  -2950,      0,   4917,  10430,
    14751,  16384,  14751,  10430,   4917,      0,  -2950,  -3477,  -2107,
        0,   1639,   2086
};
```

16 bit fixed point number with 16 bits right of the point. (2's complement!)

For example:

$$1639_d = 0000\ 0110\ 0110\ 0111_b$$

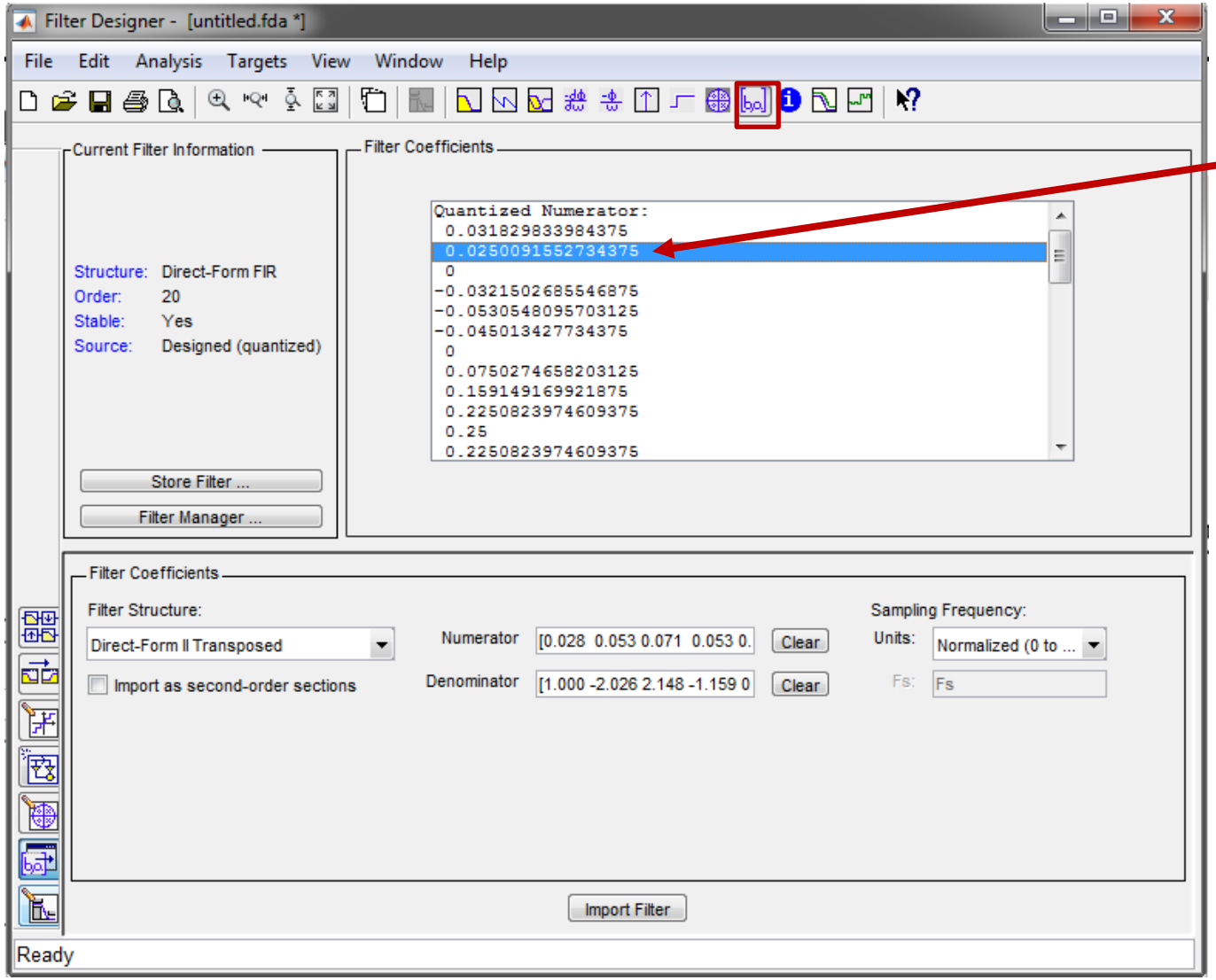
$$to\ fixed\ point:$$

$$0.0000011001100111_b$$

$$= +(2^{-6} + 2^{-7} + 2^{-10} + 2^{-11} + 2^{-14} + 2^{-15} + 2^{-16})$$

$$\approx 0.025009155_d$$

CHECK



$\approx 0.025009155_d$
Correct!

Easier:

$$\frac{1639}{2^{16}} = 0.025009155_d$$

HOGESCHOOL ROTTERDAM

MULTIPLICATION

```
   0111     =    0.875
   0001     =    0.125
   ----*         -----*
   0111           4375
  0000           1750
 0000            875
0000
-------+       ------+
0000111        109375
==>
3+3 numbers right of the point so the point is at:
0.000111 = 0.109375
```

Q0.3 notation

MULTIPLY ACCUMULATE ARITHMETIC

```
Standard integer MAC                |   Fixed point MAC
------------------------------------|-----------------------------------------
x = 0100          =  4              |   Q0.3: x = 0100 => 0.5
y = 0010          =  2              |   Q0.3: y = 0010 => 0.25
                                    |
  x  +   y * y    =  8              |   x + y * y = 0.5 + 0.25*0.25 = 0.5625
0100 + 0010*0010 =                  |   0.100 + 0.010*0.010          =
0100 + 0000100   =                  |   0.100 + 0.000100             =
                                    |
   0100                             |   0.100000
 0000100                            |   0.000100
 -------+                           |   --------+
 0001000          =  8              |   0.100100                  = 0.5625
                                    |
                                    |   Now fixed point notation is Q0.6!
```

Addition with points aligned

Add zeroes

Correct placement of the point

C-HEADER

```
/*
 * Filter Coefficients (C Source) generated by the Filter Design and Analysis Tool
 * Generated by MATLAB(R) 9.0 and the DSP System Toolbox 9.2.
 * Generated on: 15-Sep-2016 15:23:37
 */

/*
 * Discrete-Time FIR Filter (real)
 * -------------------------------
 * Filter Structure  : Direct-Form FIR
 * Filter Length     : 21
 * Stable            : Yes
 * Linear Phase      : Yes (Type 1)
 * Arithmetic        : fixed
 * Numerator         : s16,16 -> [-5.000000e-01 5.000000e-01)
 * Input             : s16,15 -> [-1 1)
 * Filter Internals  : Full Precision
 *   Output          : s33,31 -> [-2 2)  (auto determined)
 *   Product         : s31,31 -> [-5.000000e-01 5.000000e-01)  (auto determined)
 *   Accumulator     : s33,31 -> [-2 2)  (auto determined)
 *   Round Mode      : No rounding
 *   Overflow Mode   : No overflow
 */

/* General type conversion for MATLAB generated C-code  */
#include "tmwtypes.h"
/*
 * Expected path to tmwtypes.h
 * C:\Program Files\MATLAB_2016a\extern\include\tmwtypes.h
 */
const int BL = 21;
const int16_T B[21] = {
     2086,   1639,      0,  -2107,  -3477,  -2950,      0,   4917,  10430,
    14751,  16384,  14751,  10430,   4917,      0,  -2950,  -3477,  -2107,
        0,   1639,   2086
};
```

Why 31 bit right of the point?

Why 33 bits total?

How to implement this in C?

Summary

- Fixed point versus Floating point:
  - Advantages:
    - less time / chip area / power
    - fixed precision
  - Disadvantages:
    - small dynamic range (watch out for overflows)
- Fixed point calculations:
  - Multiply: use integer multiplication, remember the position of the binary point yourself:
    - $\boldsymbol{Q1.14 \cdot Q2.13 \rightarrow Q3.27}$
  - Addition: use integer addition, remember to align the binary points by shifting before adding:
    - $\boldsymbol{Q0.3 + Q1.2 \rightarrow (Q0.3 >> 1) + Q1.2 \rightarrow Q1.2 + Q1.2 \rightarrow Q2.2}$