DTP2_PROJ_5

School of Engineering

Direct Digital Synthesis (DDS)

Basic Principle

Implementation Example

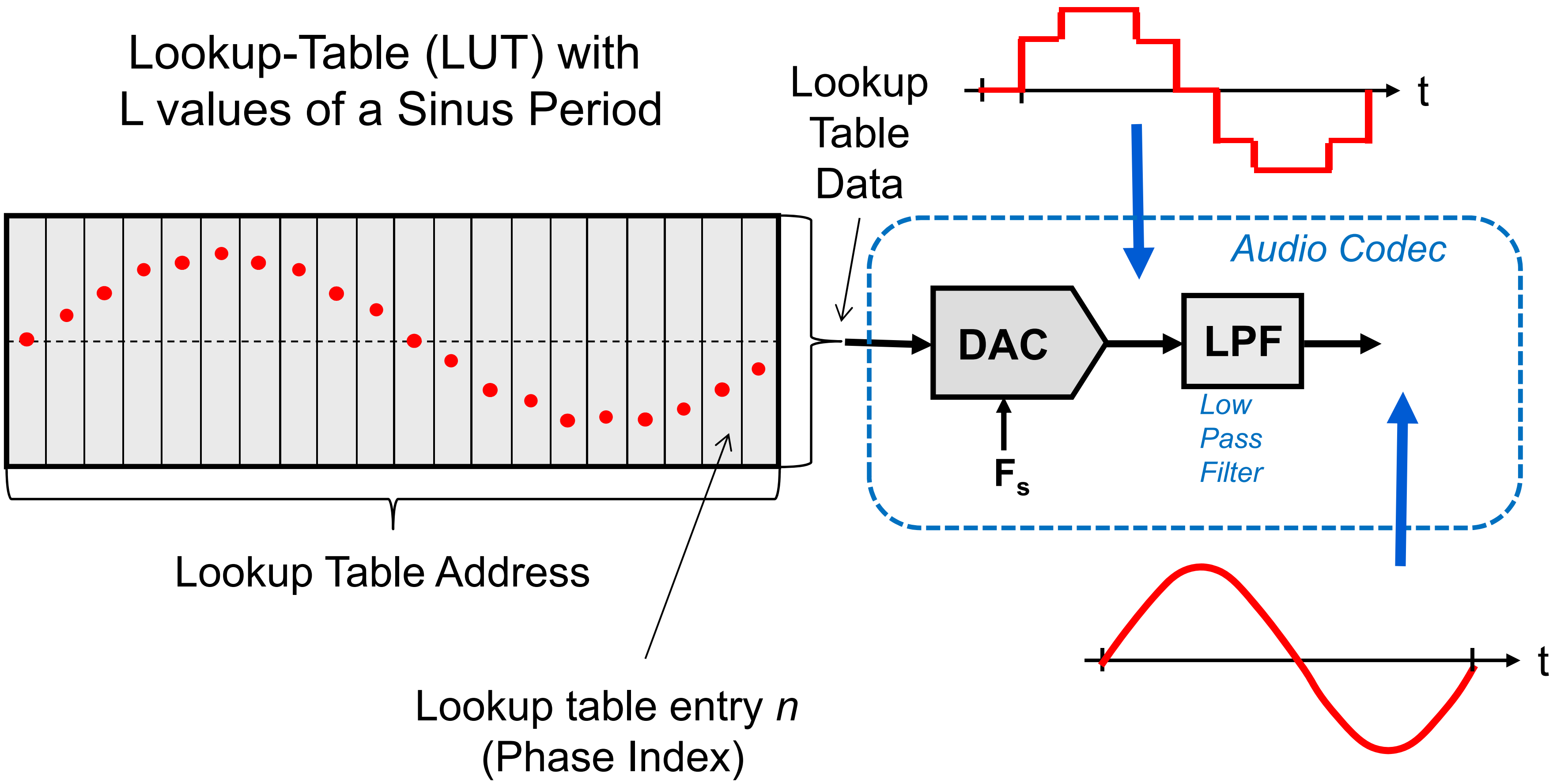
Cumulator-Length and Increment Calculation

Block Partitioning

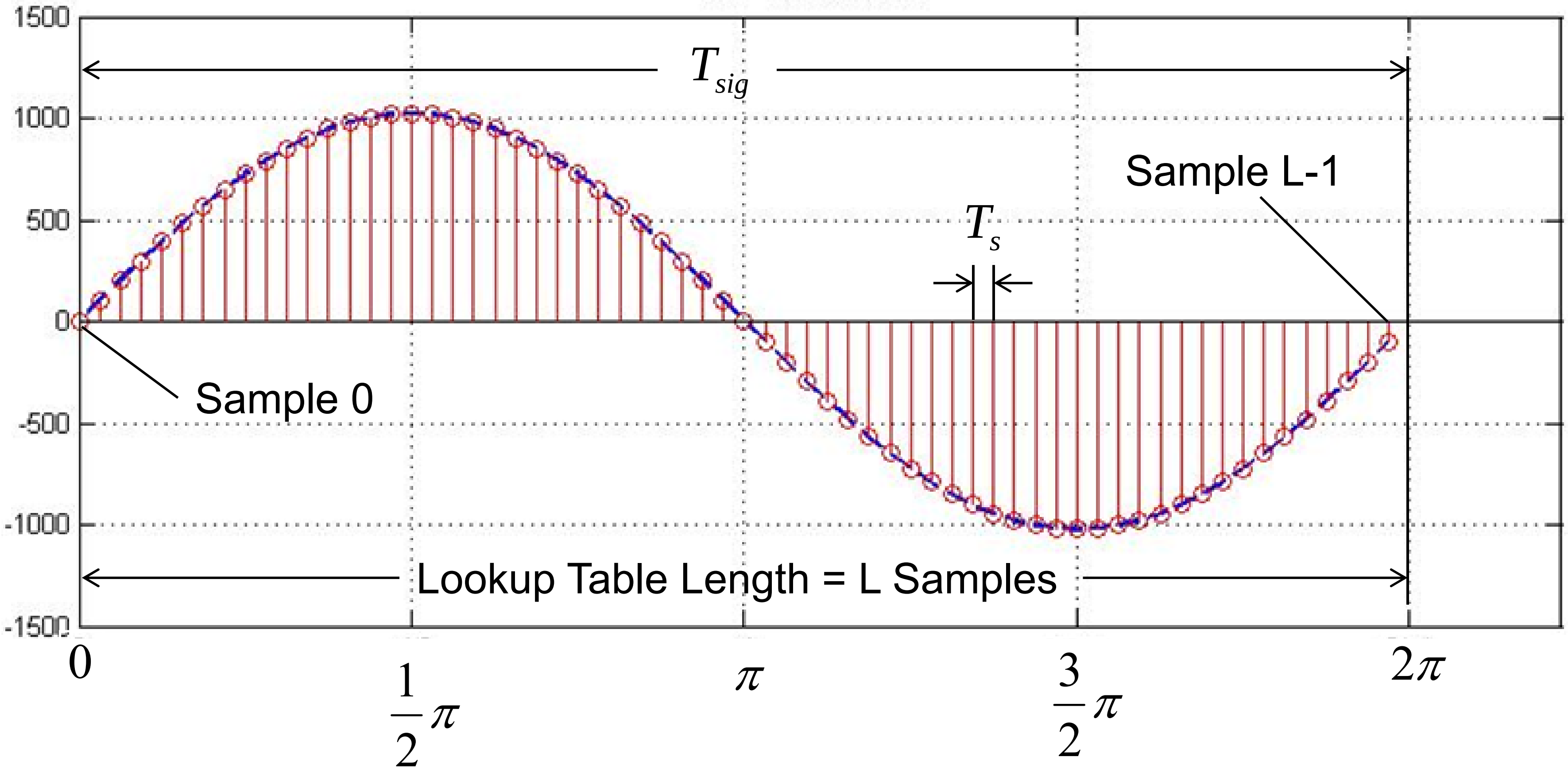
VHDL Tips

DDS: Basic Principle

School of Engineering



Synthesizing a Sinus



$$\boxed{T_{sig} = L \cdot T_s} \qquad F_s = \frac{1}{T_s} \qquad f_{sig} = \frac{1}{T_{sig}} \qquad \boxed{f_{sig} = \frac{F_s}{L}}$$

DDS: Basic Principle

School of Engineering

How to generate a sinus signal with different frequencies using :

- a counter
- a table with the values of one period of a sine wave (look-up-table : LUT)

desired signal: $\sin(\varphi) \quad mit \quad \varphi \in [0\,;2\pi]$

use the counter to increment the value of φ

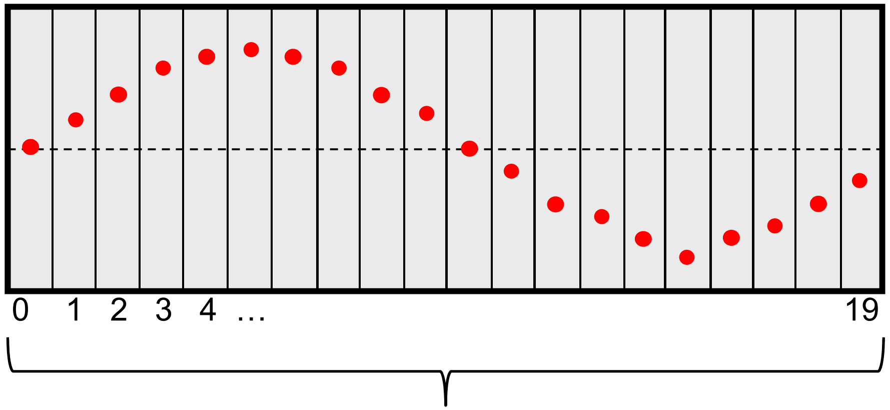
$$\varphi[k] = (\varphi[k-1] + \Delta\varphi)\bmod 2\pi$$

Then varying Δφ you can vary the frequency of the sine wave

DDS: Basic Principle

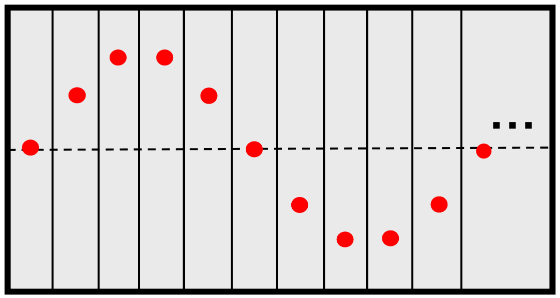
Consider the LUT below with L values (L=20).

The values stored in the LUT correspond to $\rightarrow \sin\left(\frac{2\pi}{20}\cdot n\right) \quad mit \quad n \in [0\,;19]$

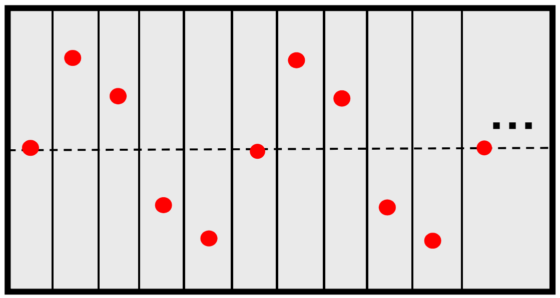


n = Lookup Table Address

using M=Δn=2



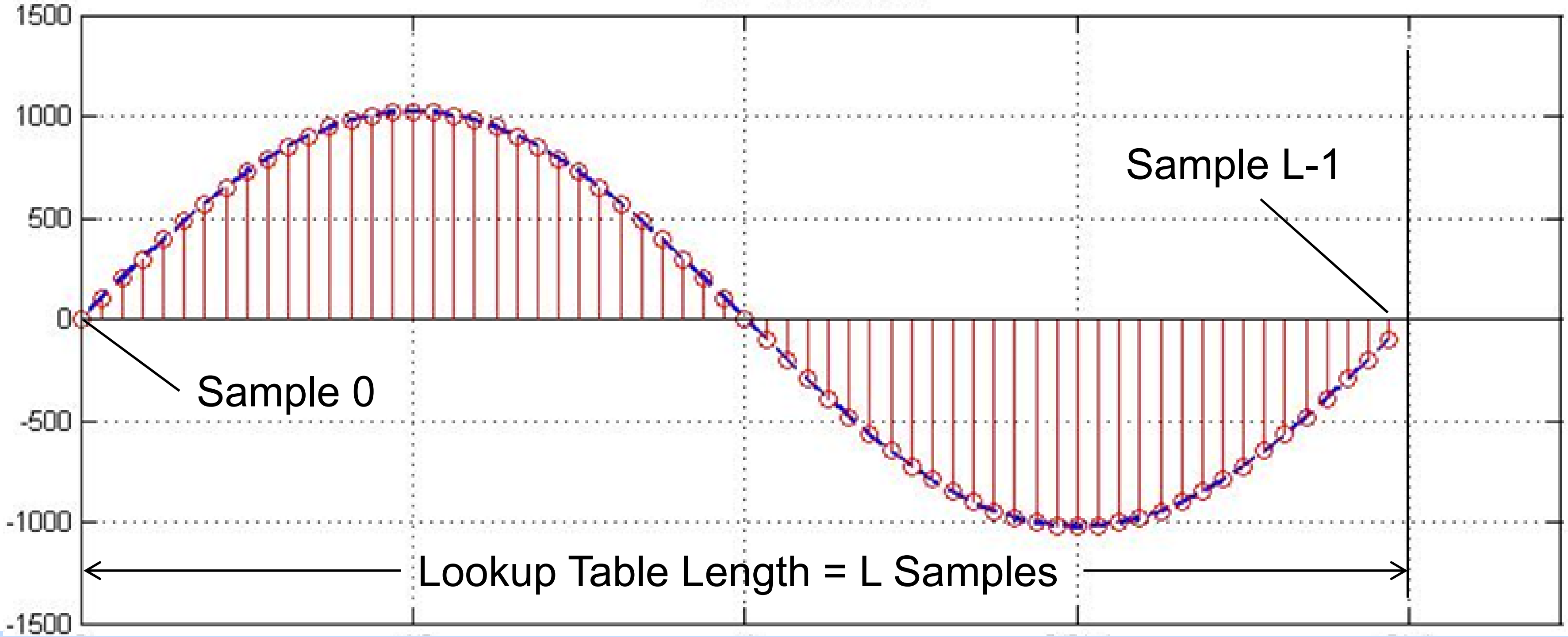
using M=Δn=4



The counter increments the value of n, and let us call the increment Δn = M

$$n[k] = \left(n[k-1] + \Delta n \right) \bmod L$$

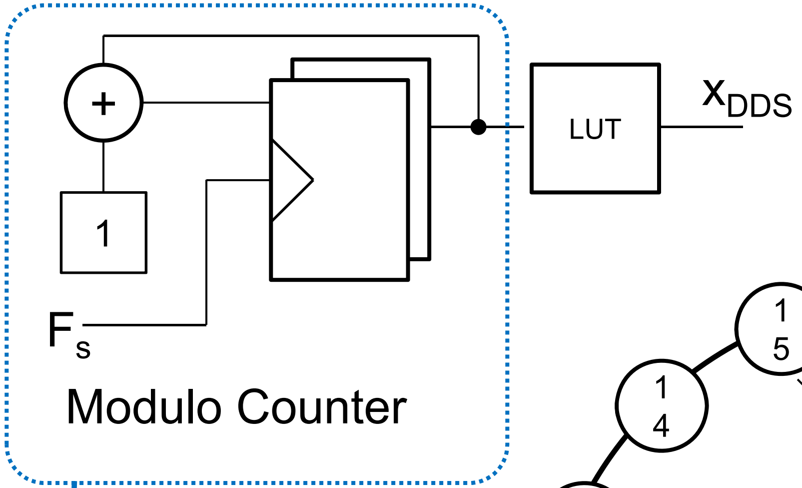
Sinus Look-up-Table (LUT)



index:	0	L/4	L/2	3/4L	L
phase: (equiv)	0	π/2	π	3π/2	2π

LUT Adress Generation with Modulo Counter

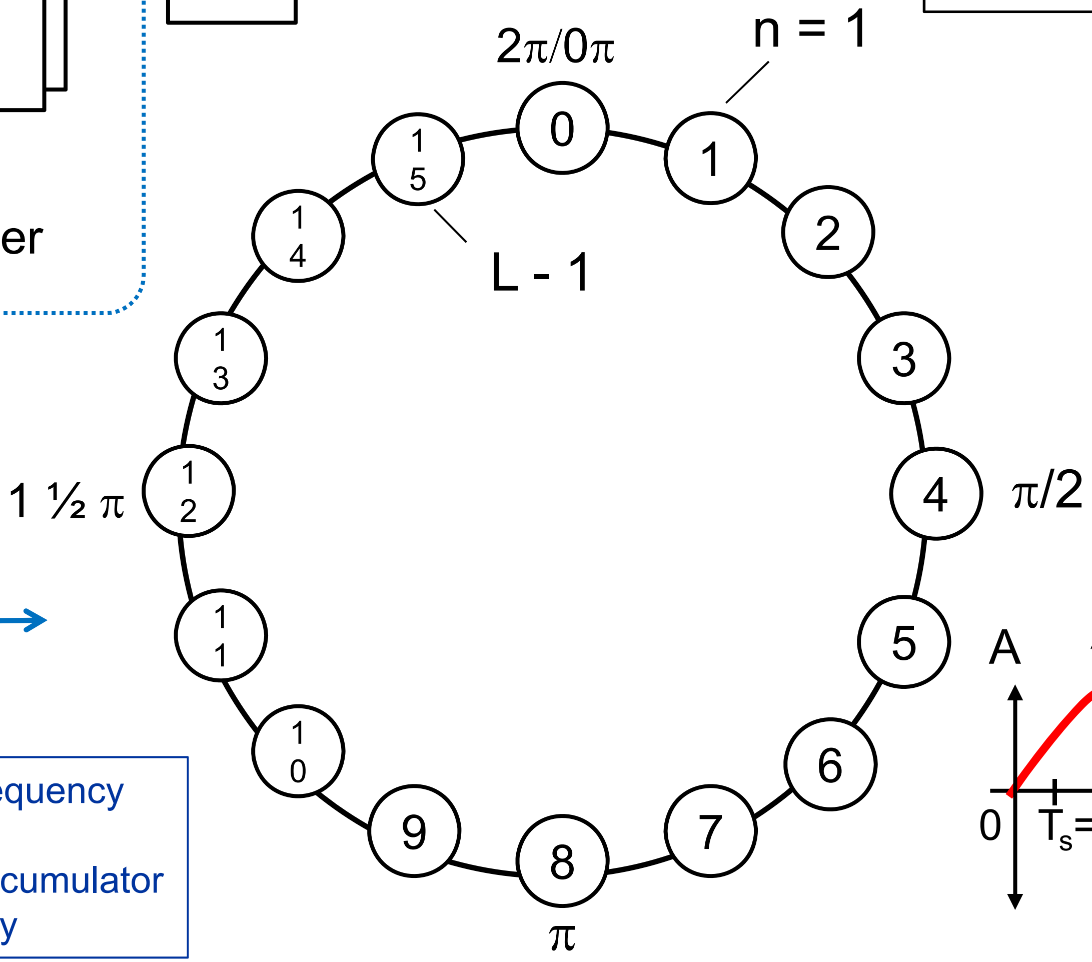
School of Engineering

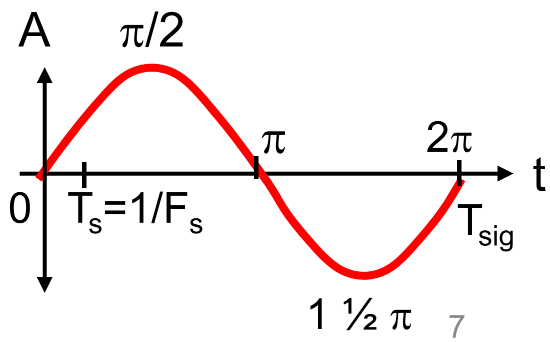


$$x_{DDS}[n] = A \cdot \sin\left(n \cdot \frac{2\pi}{L}\right)$$

$$f_{sig} = \frac{Fs}{L}$$

example
with L = 16





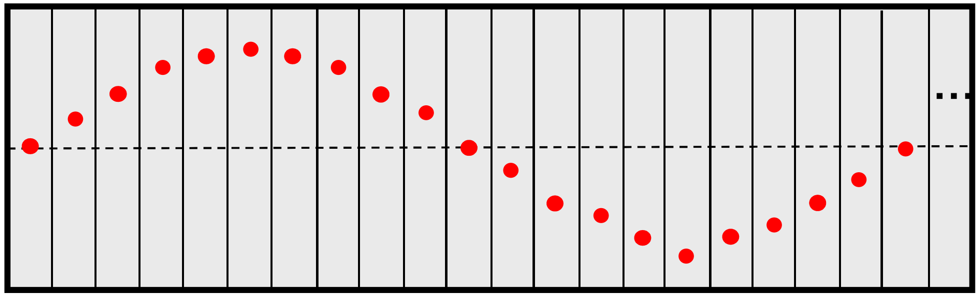
F_s= audio sample frequency
L = length of LUT
n = index for phase cumulator
f_{sig} = signal frequency

Example when LUT Length L=20

School of Engineering

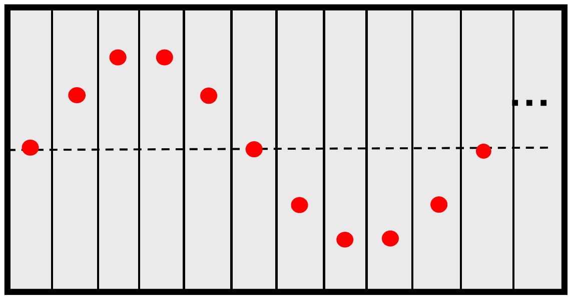
The counter increment value ($M=\Delta n$) can be varied!

using $M=\Delta n=1$



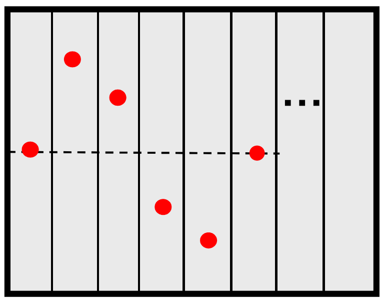
Lookup Table Address n = 0 1 2 3 4 … 19 0 …

using $M=\Delta n=2$



Lookup Table Address n = 0 2 4 6 4 18 0 …

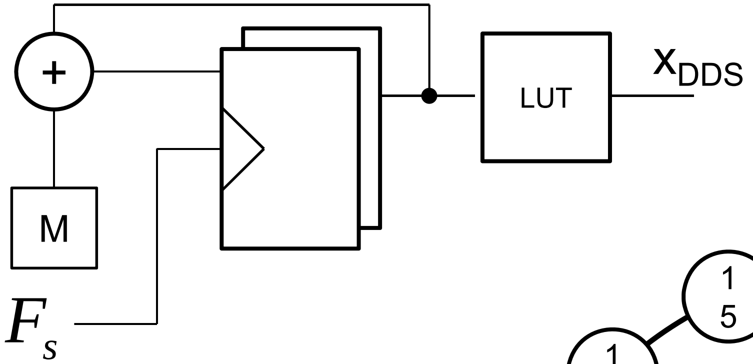
using $M=\Delta n=4$

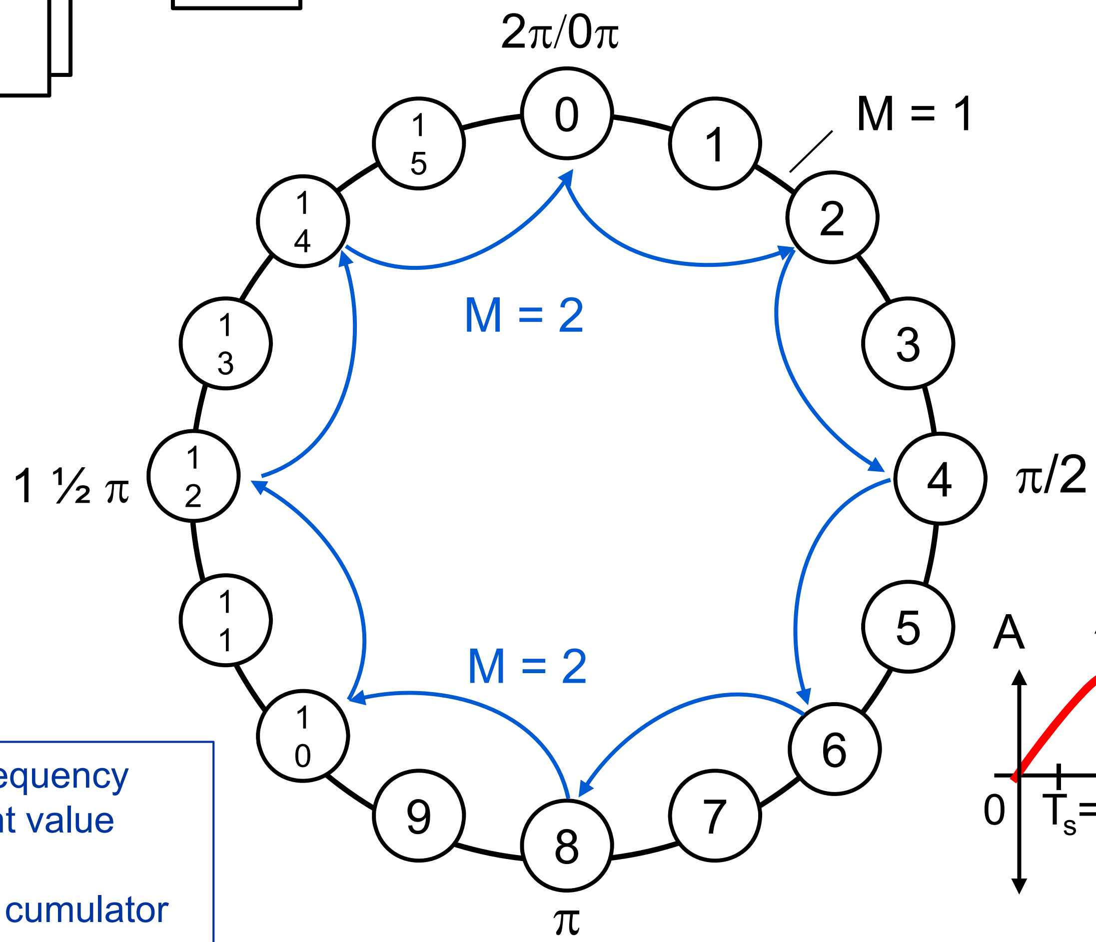


Lookup Table Address n = 0 4 8 12 16 0 …

Controlling Frequency of f_{sig}

School of Engineering

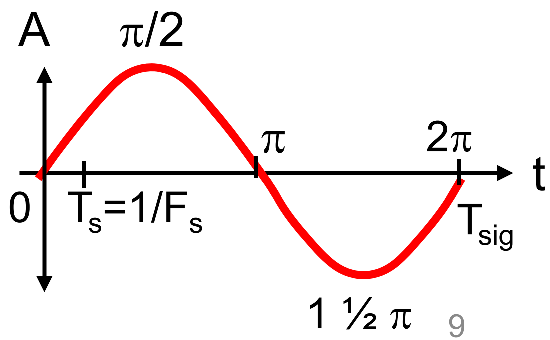




$$f_{sig} = \frac{F_s}{L} \cdot M$$

or to calculate M

$$M = \frac{L}{F_s} \cdot f_{sig}$$



F_s = audio sample frequency
M = phase increment value
L = length of LUT
n = index for phase cumulator
f_{sig} = signal frequency

Controlling Frequency of f_{sig}

How to choose the value of M for a desired fsig (frequency of the generated sine wave)?

- Analog sinus signal x(t) and digital sinus signal x[n] (approx with time step T_s)

$$x(t) = A \cdot \sin(2\pi \cdot f_{sig} \cdot t) \quad \approx \quad A \cdot \sin\left(2\pi \cdot f_{sig} \cdot \frac{n}{F_s}\right) = A \cdot \sin\left(n \cdot 2\pi \cdot \frac{f_{sig}}{F_s}\right) = x[n]$$

- Comparing with digital sinus signal we can generate with DDS, and calculating M for a desired f_{sig} value

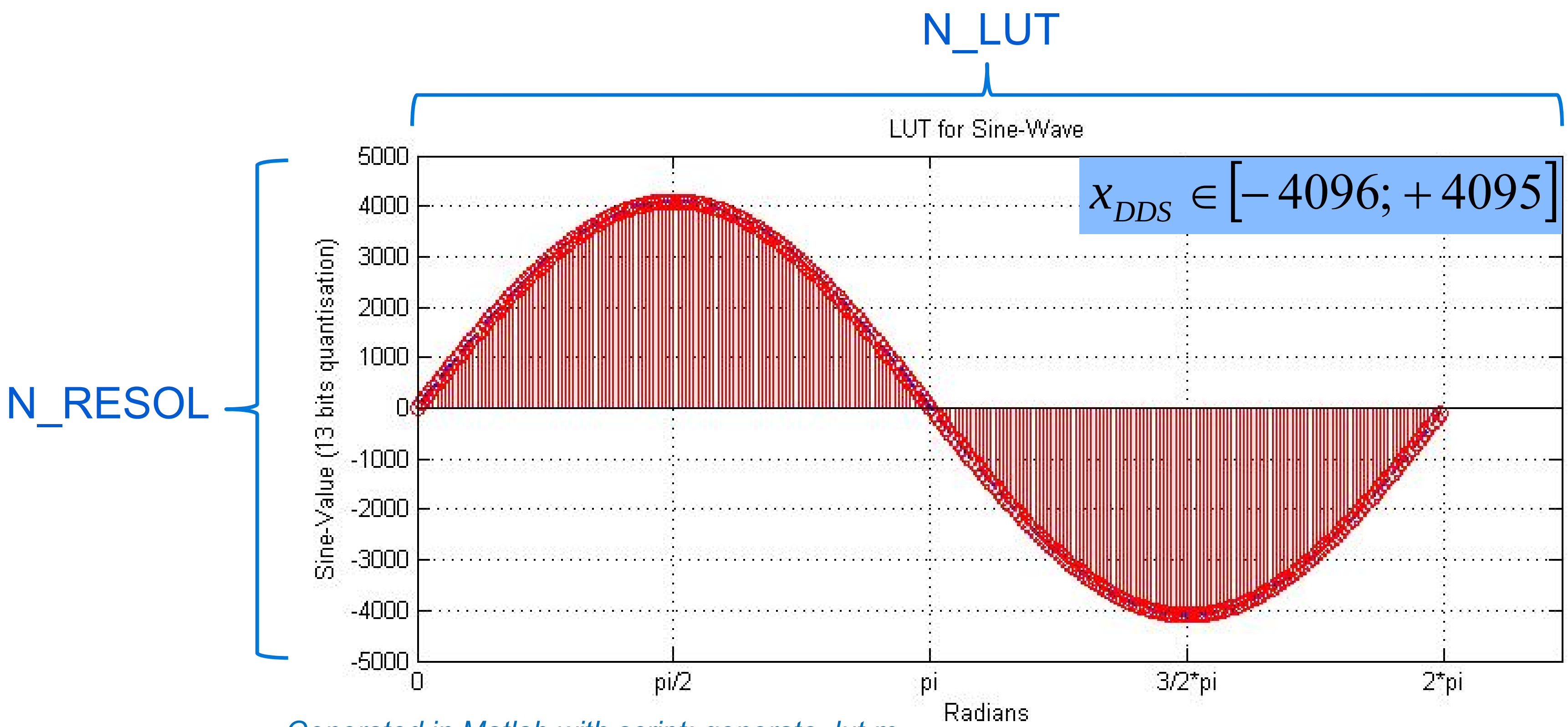
$$x[n] = A \cdot \sin\left(n \cdot 2\pi \cdot \frac{f_{sig}}{F_s}\right) \approx A \cdot \sin\left(n \cdot 2\pi \cdot \frac{M}{L}\right) = x_{DDS}[n]$$

$$\Rightarrow \qquad M \quad \approx \quad L \cdot \frac{f_{sig}}{F_s} \qquad with \quad M \in Z^{+}$$

Proposal for LUT Implementation

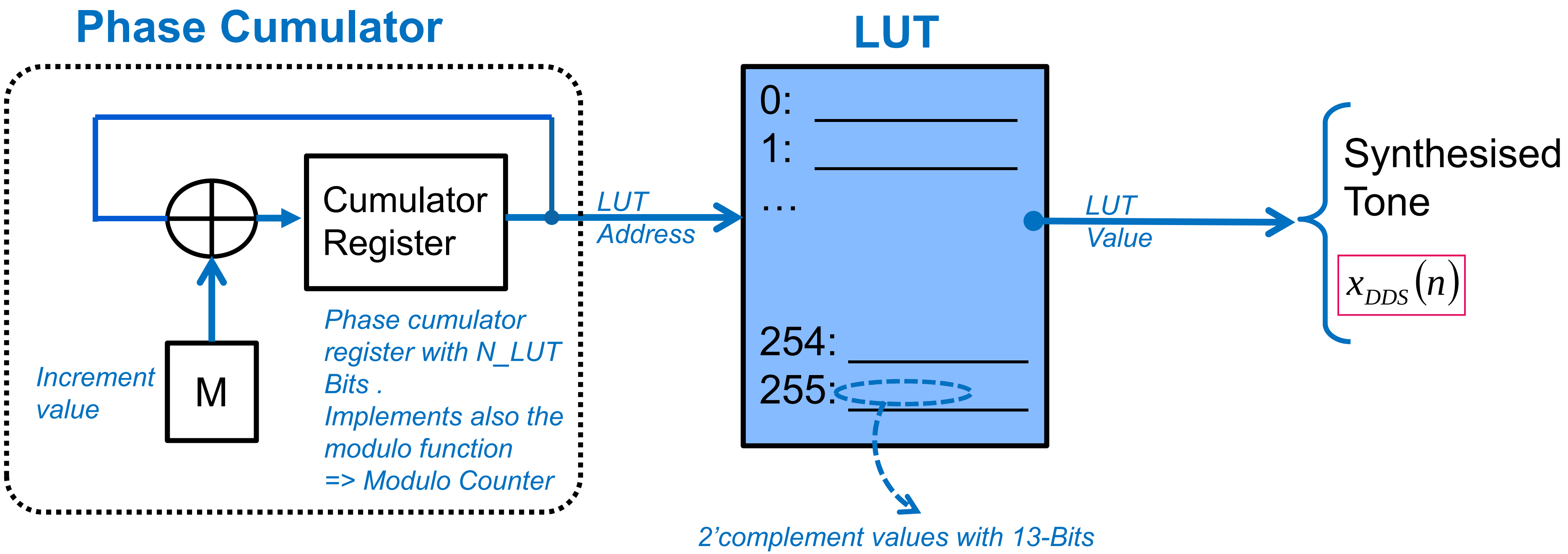
School of Engineering

N_LUT = 8 bits	*number of bits to address LUT*
L = 2^8	*length of LUT*
N_RESOL = 13 bits	*resolution of values stored in LUT (values are in 2'complement format)*



Generated in Matlab with script: generate_lut.m

Resulting Frequency Steps



Which frequency is synthesised with update on phase cumulator every F_s= 48kHz and:

$$f_{sig} = \frac{F_s}{L} \cdot M = \frac{48k}{256} \cdot M$$

- M=1 ... $\Rightarrow f_{sig} = 187{,}5Hz$
- M=2 ... $\Rightarrow f_{sig} = 375Hz$
- M=3 ... $\Rightarrow f_{sig} = 562{,}5Hz$

DDS: Implementation Example

School of Engineering

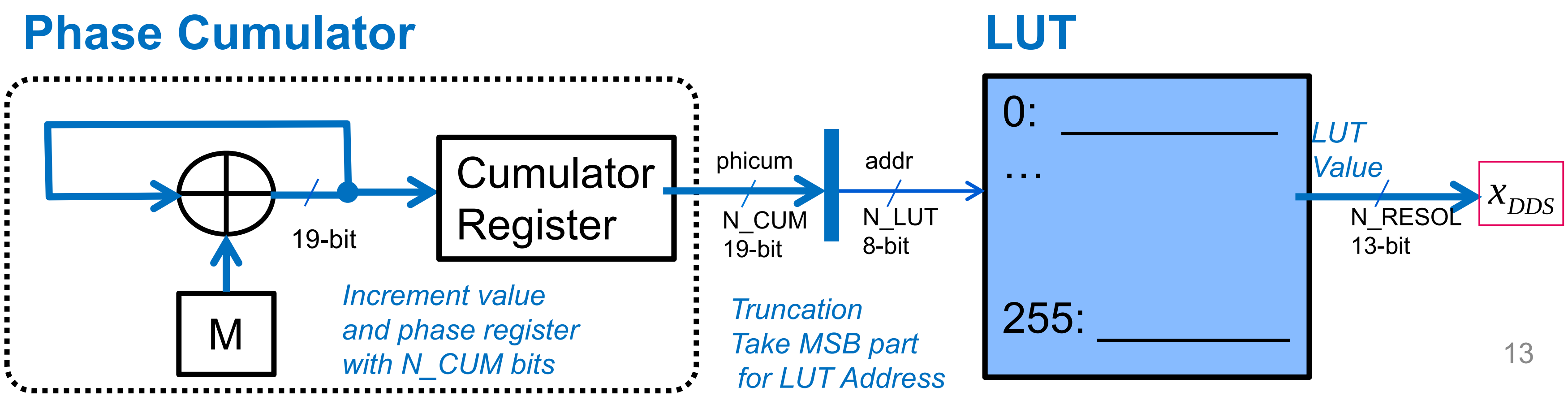
So far with integer values of M, we can get a minimum frequency step of:

$$\Delta f = \frac{F_s}{2^{N\_LUT}} = \frac{F_s}{L} = \frac{48k}{256} = 187{,}5Hz$$

How can one improve the precision of the signal frequency? (get a finer freq. step)

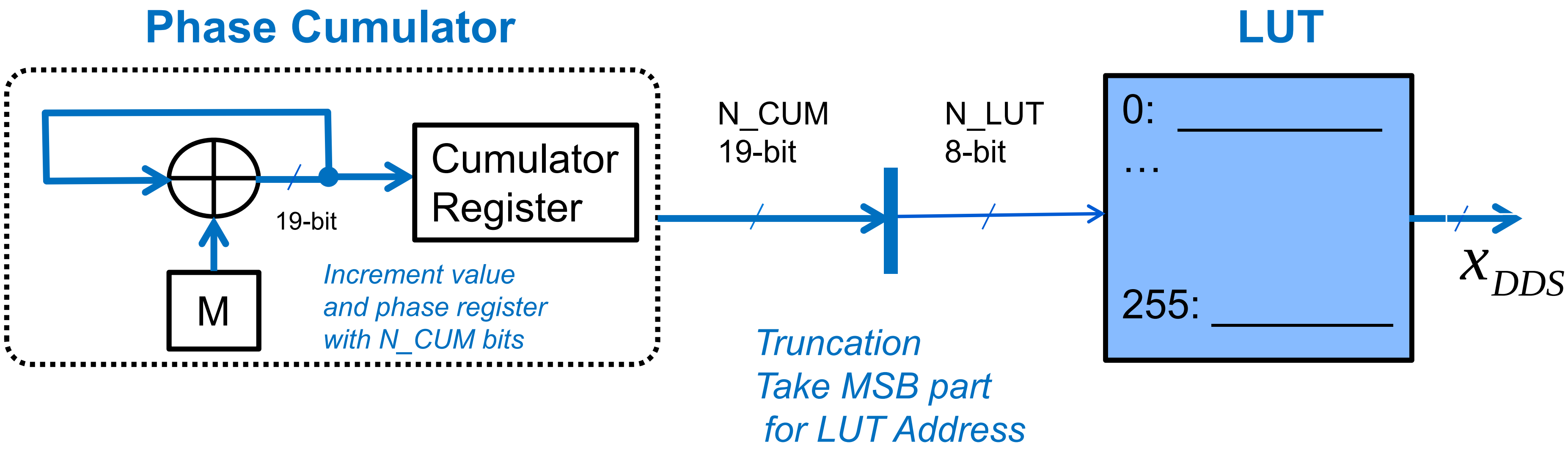
By using a fractional M value (with some bits after the comma).

This is equivalent to counting M with N_CUM bits (N_CUM > N_LUT), but taking only the MSB part (N_LUT bits) to address the LUT.



Making Counter Bits > N_LUT

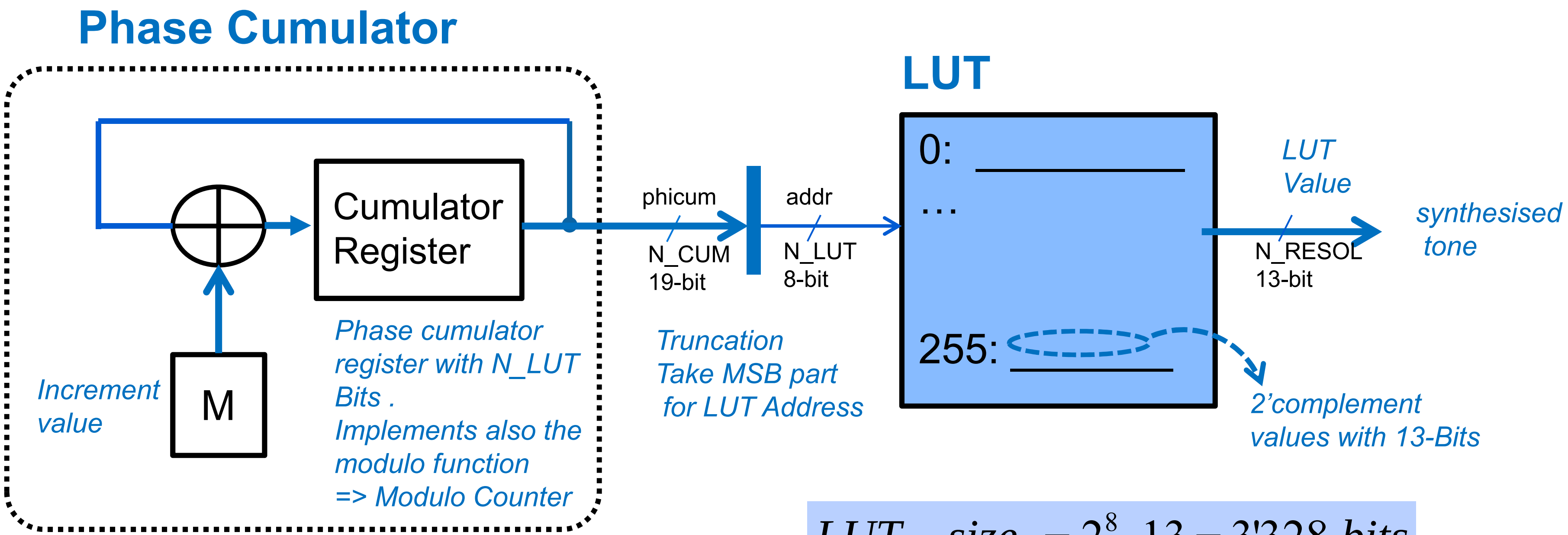
School of Engineering



$$\Delta f = \frac{F_s}{2^{N\_CUM}} = \frac{48k}{2^{19}} = 0{,}0916 Hz$$

Frequency steps when N_LUT = 19 bits

School of Engineering



$$LUT\_size = 2^8 \cdot 13 = 3'328\ bits$$

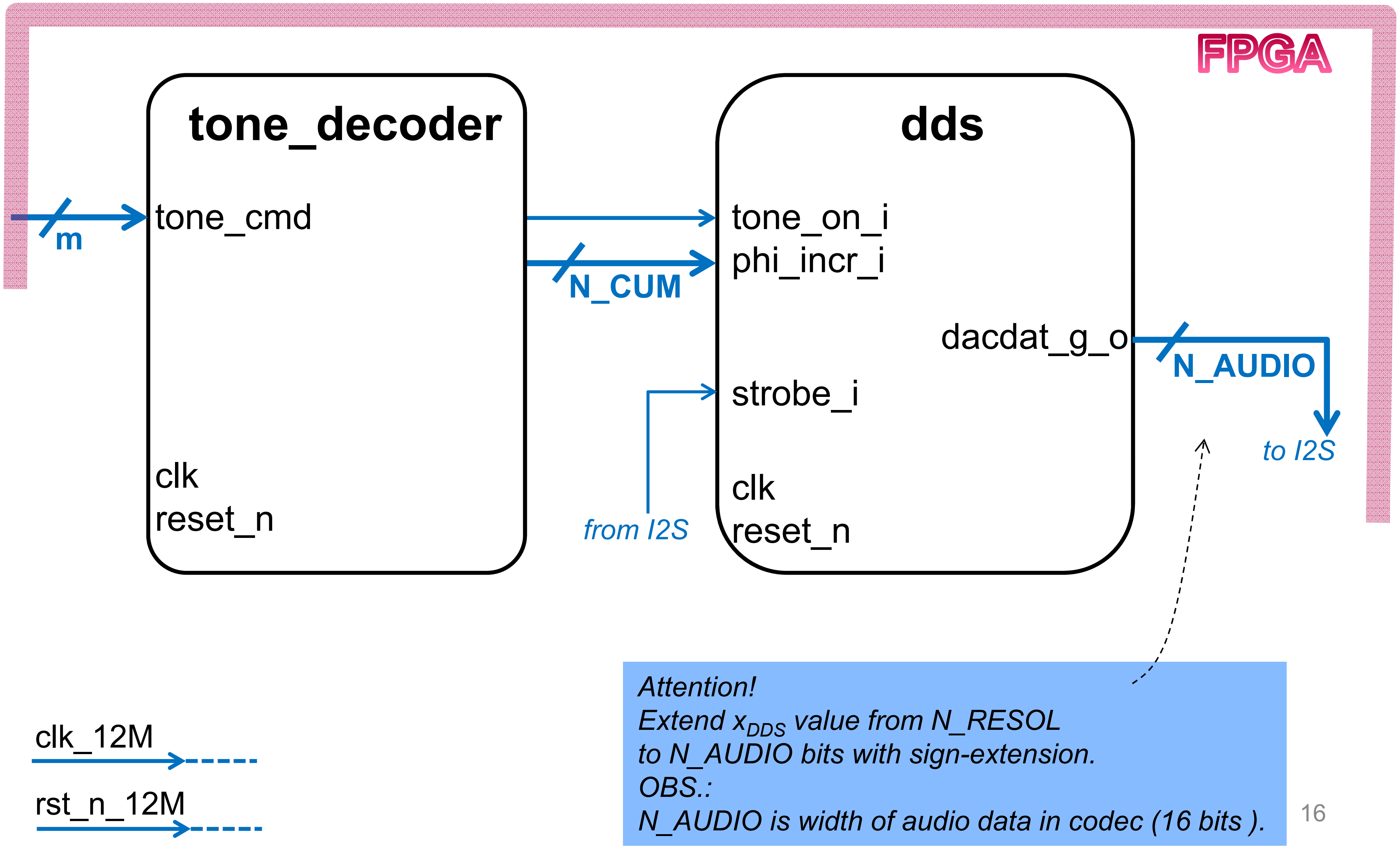
Available Memory Bits in 4CE-115 FPGA = 3'888'000 bits

Which frequency is synthesised when phase cumulator works with F_s= 48kHz and N_LUT = 19 bits?

$$f_{sig} = \frac{f_s}{L} \cdot M = \frac{48k}{2^{19}} \cdot M$$

- M=1 ... $\Rightarrow f_{sig} = 0{,}092 Hz$
- M=2 ... $\Rightarrow f_{sig} = 0.183 Hz$

Tone Generator



Attention!
Extend x_{DDS} value from N_RESOL
to N_AUDIO bits with sign-extension.
OBS.:
N_AUDIO is width of audio data in codec (16 bits).

DDS VHDL Code – Tips 1/3

School of Engineering

- Use the strobe signal from i2s_master to enable the update of phase cumulator;

  The signal strobe should have 1 pulse every cycle of 1/48kHz

- Package **tone_gen_pkg.vhd** contains definition of constants plus a type for the declaration of the LUT

```
-------------------------------------------------------------------------------
 -- Constant Declaration
-------------------------------------------------------------------------------
   CONSTANT N_CUM:   natural :=19;           -- number of bits in phase cumulator
   CONSTANT N_LUT:   natural :=8;            -- number of bits in LUT address
   CONSTANT L:       natural := 2**N_LUT;    -- length of LUT
   CONSTANT N_RESOL: natural := 13;          -- Attention:1 bit reserved for sign
-------------------------------------------------------------------------------
-- Type Declaration
-------------------------------------------------------------------------------
   SUBTYPE t_audio_range IS integer RANGE-(2**(N_RESOL-1)) TO (2**(N_RESOL-1))-1;
                                               -- range : [-2^12; +(2^12)-1]
   TYPE t_lut_rom IS ARRAY(0 to L-1) OF t_audio_range;
   CONSTANT LUT : t_lut_rom :=(0,101,201,301,401,501,601,700,799,897,995, … );
```

DDS VHDL Code – Tips 2/3

School of Engineering

- Syntax to grab N_LUT MSBs and use as address to search value in LUT
  remember to convert address to integer before using it as index!

```
CONSTANT N_AUDIO:  natural :=16;     --width audio data in codec
…
SIGNAL phicum_reg: unsigned(N_CUM-1 downto 0);
SIGNAL addr :      integer range 0 to L-1;

...
-- take N_LUT  MSBs as address to select value in LUT
addr <= to_integer( phicum_reg (N_CUM-1  DOWNTO N_CUM - N_LUT) );

...
-- convert to signed with 16 bits to match settings in codec for audio data resolution
-- conversion with to_signed takes care of sign extension
dacdat_g_o <= std_logic_vector(to_signed (LUT(addr),N_AUDIO ) );
```

DDS VHDL Code – Tips 3/3

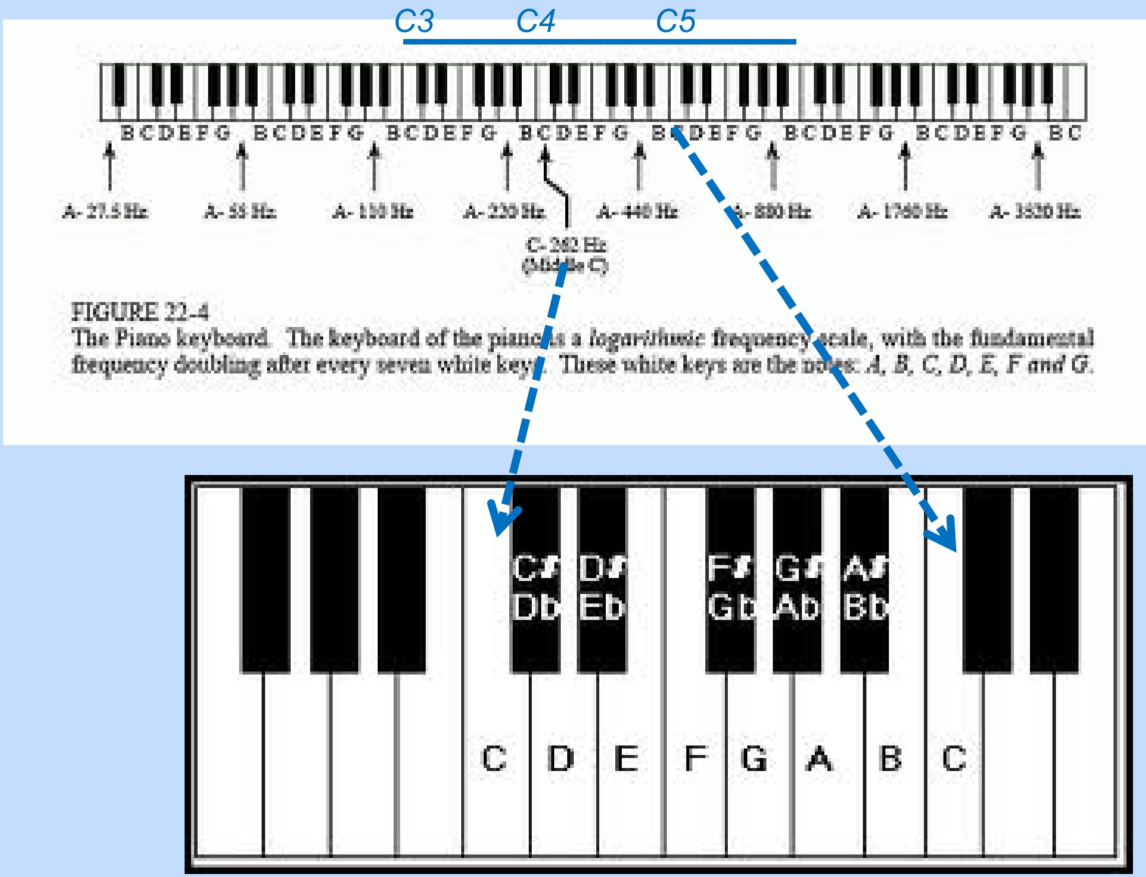
- Increment values are pre-calculated and coded as constants in **tone_gen_pkg.vhd**

-- Piano Mid-Octave (white keys)

-- DO-C4 tone ~261.63Hz
-- RE_D4 tone ~293.66Hz
-- MI_E4 tone ~329.63Hz
-- FA_F4 tone ~349.23Hz
-- SOL_G4 tone ~392.00Hz
-- LA_A4 tone ~440.00Hz
-- SI_B4 tone ~493.88Hz
-- DO_C5 tone ~523.25Hz

-- Piano Mid-Octave (black keys)

-- DOS_C4S tone ~277.18Hz
-- RES_D4S tone ~311.13Hz
-- FAS_F4S tone ~369.99Hz
-- SOLS_G4S tone ~415.30Hz
-- LAS_A4S tone ~466.16Hz



Example:

```
CONSTANT M_LA_A4: unsigned(N_CUM-1 downto 0):= to_unsigned(4806, N_CUM));
-- LA_A4  tone ~440.00Hz
-- M = 2^19 * 440/48000
```