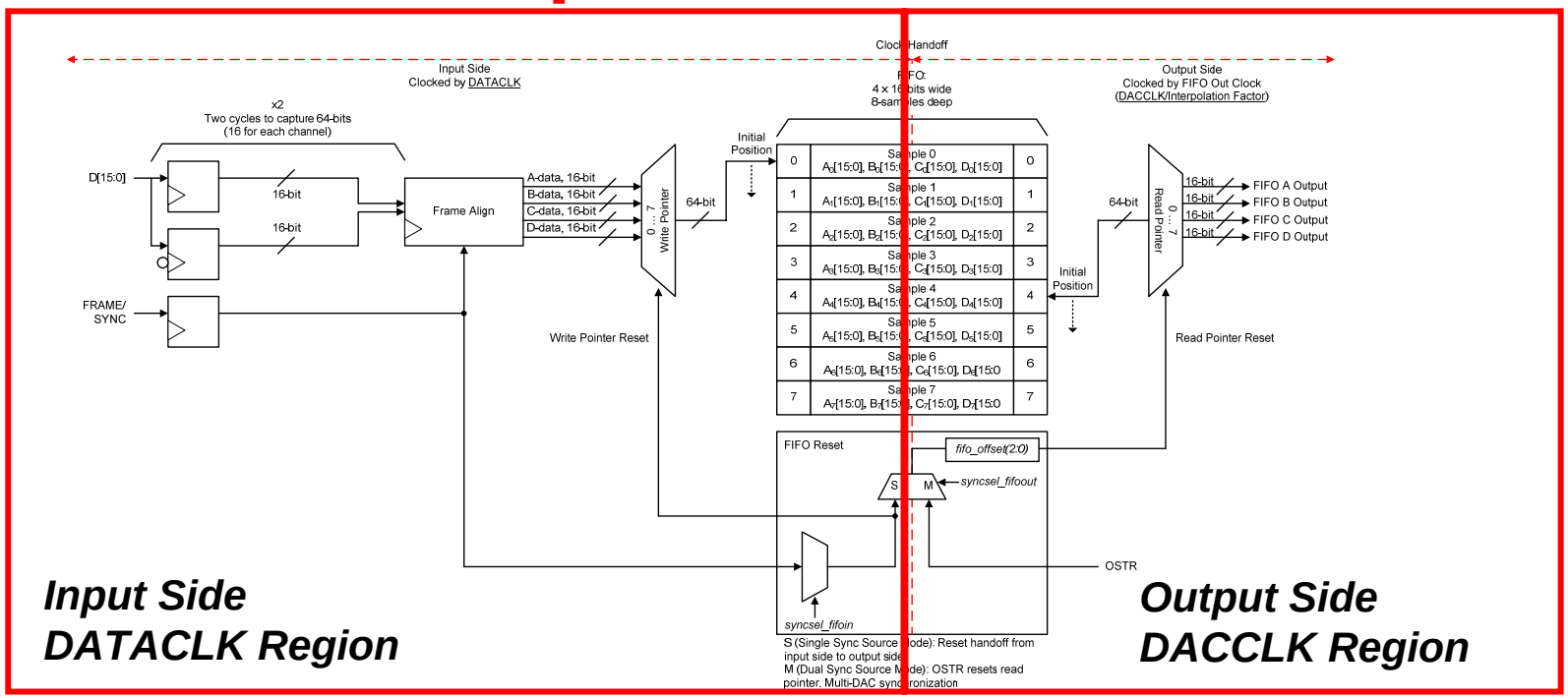
DAC3484 Multi-DAC Synchronization

Bao Nguyen/Kang Hsia

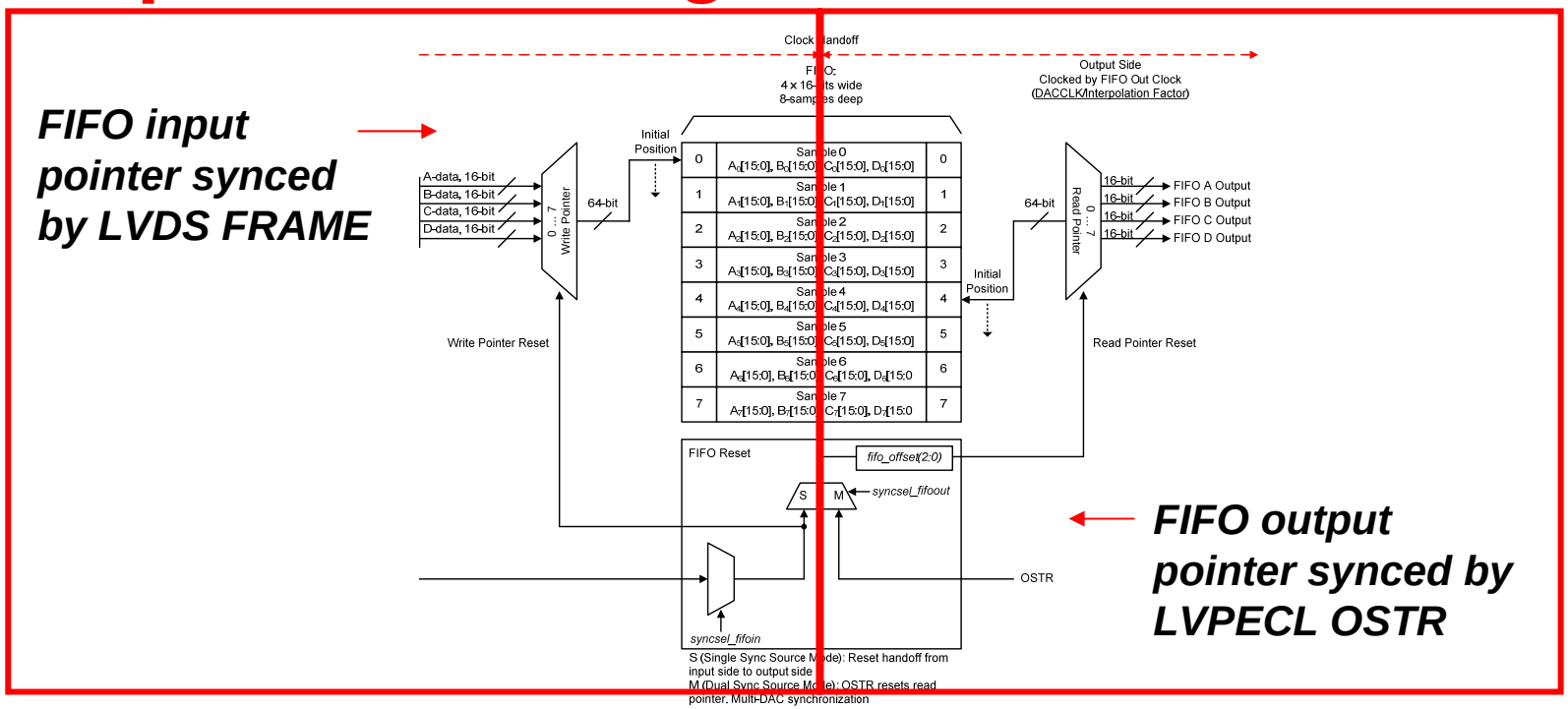
2/16/2011

DAC3484 FIFO Operation



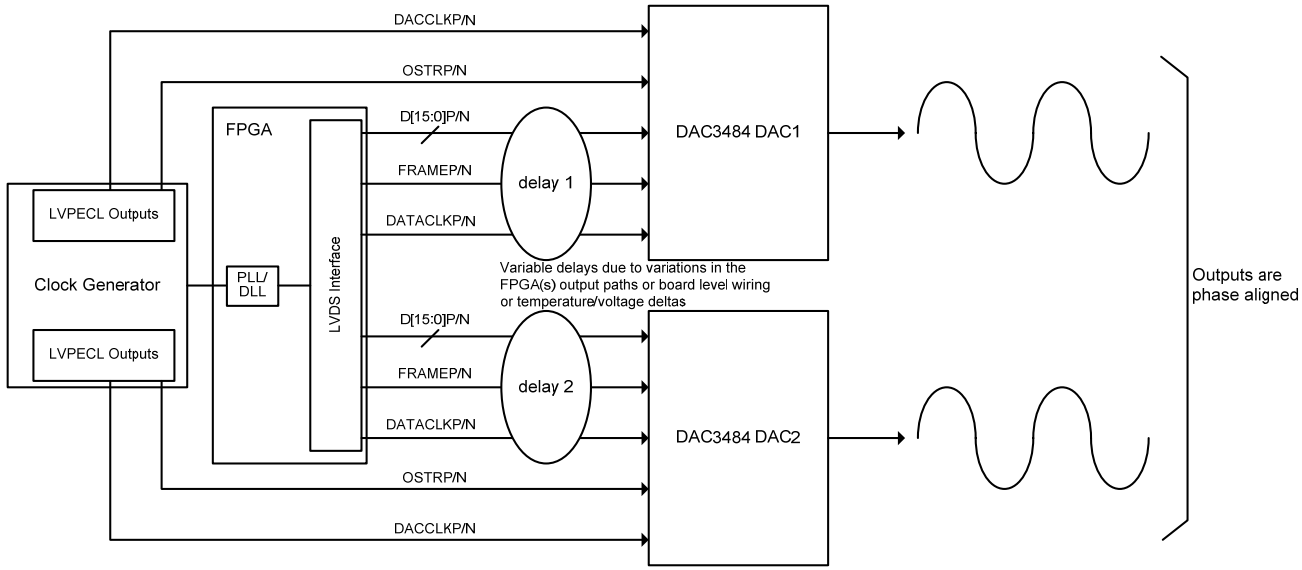
- DAC3484 has a 8-samples deep FIFO to absorb the timing variation between the input data rate and the internal DAC sample rate.
- The FIFO input write pointer and output read pointer can be synchronized by two independent sources: FRAME and OSTR, respectively. This is the Dual Sync Source Mode.

Concept of OSTR Signal



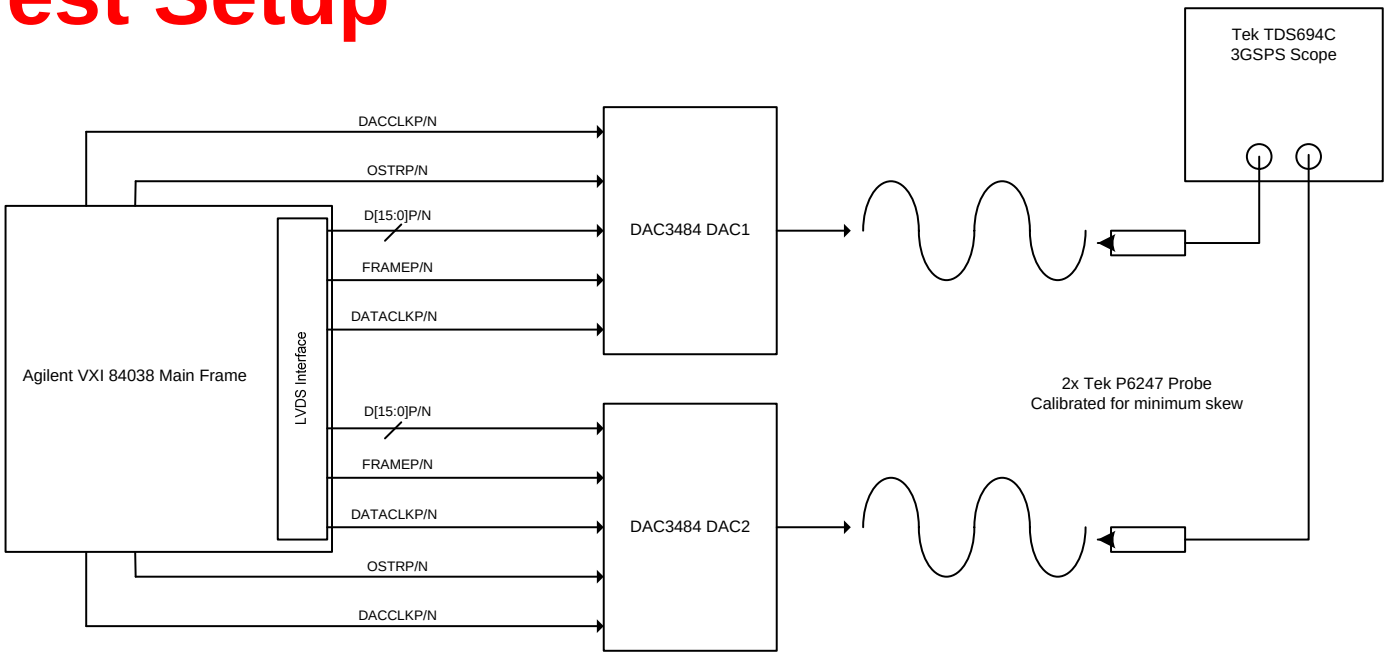
- When the DAC3484's FIFO is configured in Dual Sync Source Mode, the FIFO output pointer is synced by OSTR signal.
- The OSTR signal comes from the same source as the DACCLK. Therefore, the phase relationship between the OSTR and DACCLK are maintained by the same source.

Theory of Multi-DAC Sync



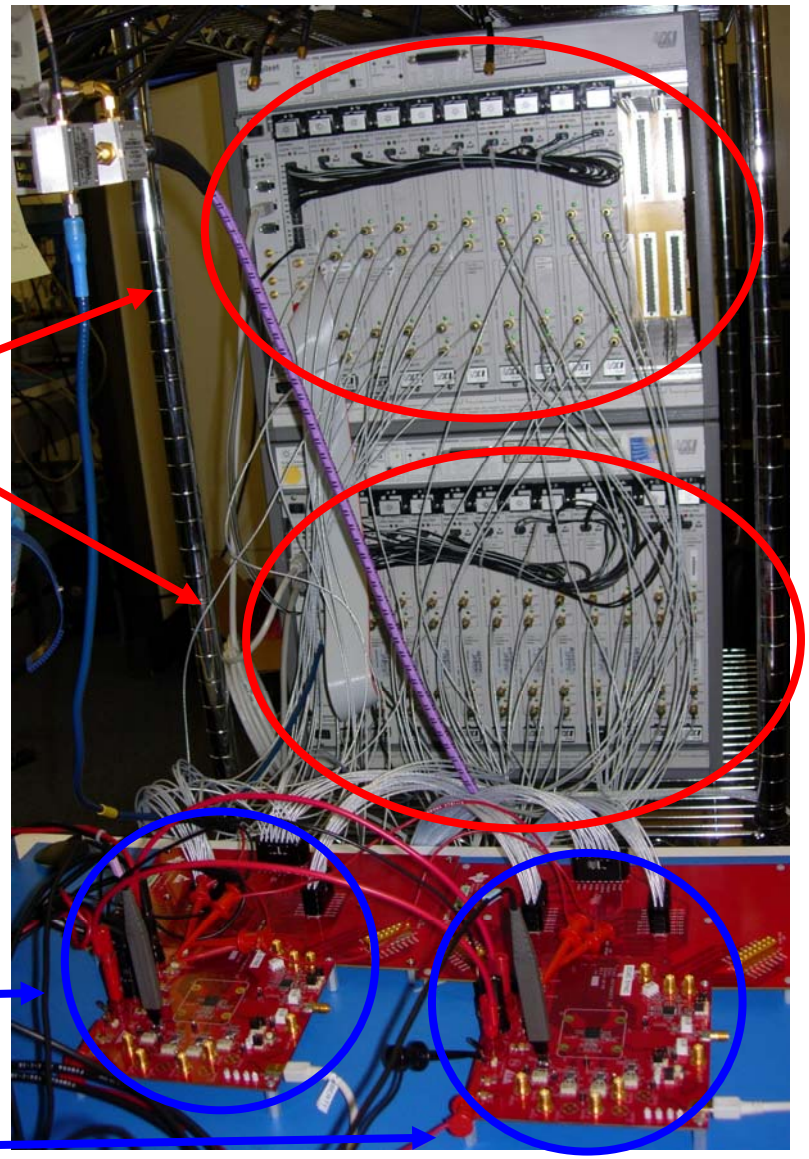
- Some applications would require multiple DACs in a system to have fully synchronized outputs.
- The outputs of multiple devices can be synchronized by configuring the following:
  - All the DAC3484 are configured in Dual Sync Source Mode
  - All the OSTR signals to the individual DAC3484 have little or no skew.
- When the above two points are achieved, then all the outputs of the multiple devices will refer to the same FIFO space when the FIFOs of multiple devices are synced by zero skew OSTR signals.
- Multiple DAC3484 devices have now the same latencies. The input data latency variation is transparent in this mode.

Test Setup



- DAC3484 Rev. B Lab EVM (2x)
- Agilent VXI 84038 Main Frame (2x).
- Each DAC3484 EVM receives DATA and CLK from the VXI
  - 16 bit LVDS lanes for DATA
  - 1x DATACLK LVDS lane
  - 1x FRAME LVDS lane
  - 1x OSTR LVPECL lane
  - 1x DACCLK LVPECL lane
  - Total of 40 DATA + CLK Lanes for each DAC3484
- Fdac = 737.28MSPS, 4x interpolation, CMIX bypassed or CMIX fs/8 enabled depending on the IF output.

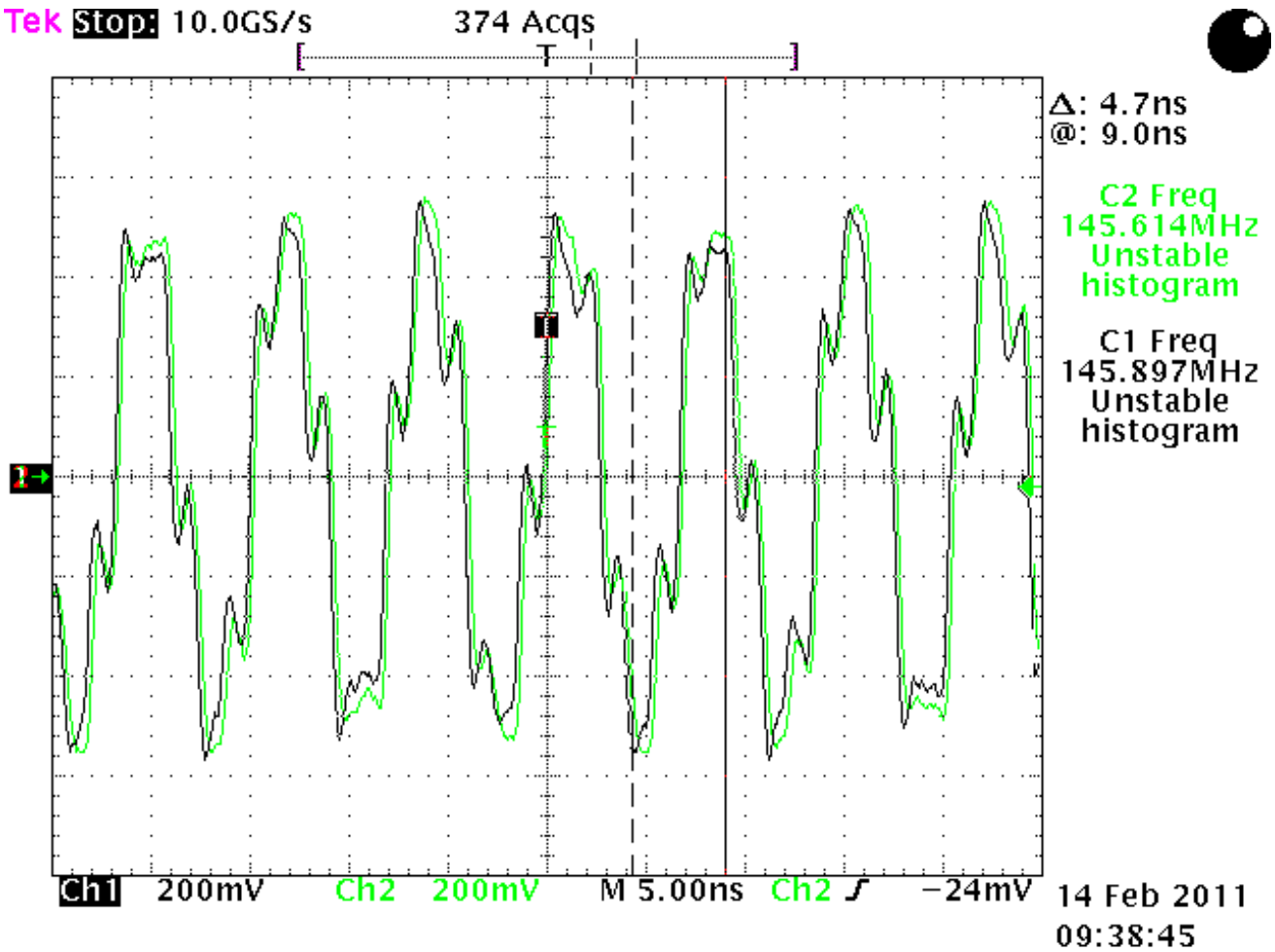
Test Setup



Part 1
Synchronization Procedure (baseband or CMIX only)

1. Set TXENABLE to logic LOW
2. Supply all DAC power supplies
3. Provide all LVPECL inputs: DACCLKP/N and OSTRP/N
4. Toggle the RESETB pin for a minimum 25ns active low pulse width
5. Program the SIF Registers to configuration the DAC3484
    1. Config31, bit<3:2> syncsel_dataformmatter = “00” to sync the FIFO data formmatter with LVDS FRAME.
    2. Config32, bit<15:12> syncsel_fifoin = “0010” to sync the FIFO input pointer with LVDS FRAME
    3. Config32, bit<11:8> syncsel_fifoout = “0100” to sync the FIFO output pointer with LVPECL OSTR
    4. Config32, bit<0> clkdiv_sync_sel = “0” to point clock divider sync source to LVPECL OSTR
    5. Config0, bit<2> clkdiv_sync_ena = “1” to enable the clock divider sync to sychronize the phase of the internal divided-down clocks
6. Provide all LVDS inputs (D[15:0]P/N, DATACLKP/N, FRAMEP/N, SYNCP/N, and PARITYP/N) simultaneously.
7. The FIFO input/output pointers and clock divider should be synced at this point by rising edge of FRAMEP/N and OSTRP/N.
8. Disable clock divider sync: Config0, bit<2> = “0”
9. Enable the data transmission by asserting the TXENABLE to logic HIGH or sif_txenable = “1”.

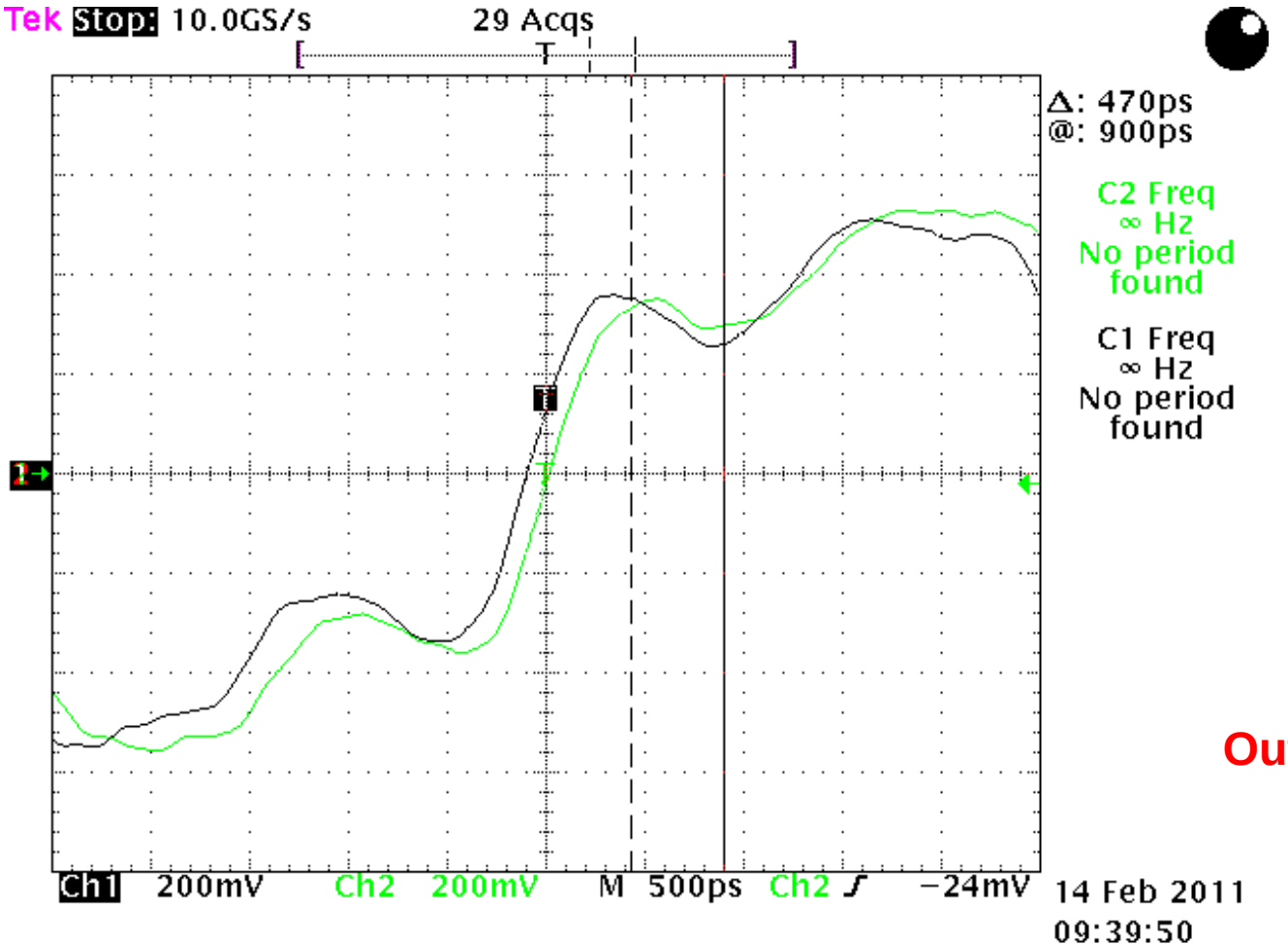
IF=140MHz, Baseband = 47.84MHz, fs/8 CMIX



Green: EVM#1

Black: EVM#2

IF=140MHz, Baseband = 47.84MHz, fs/8 CMIX (zoomed in)

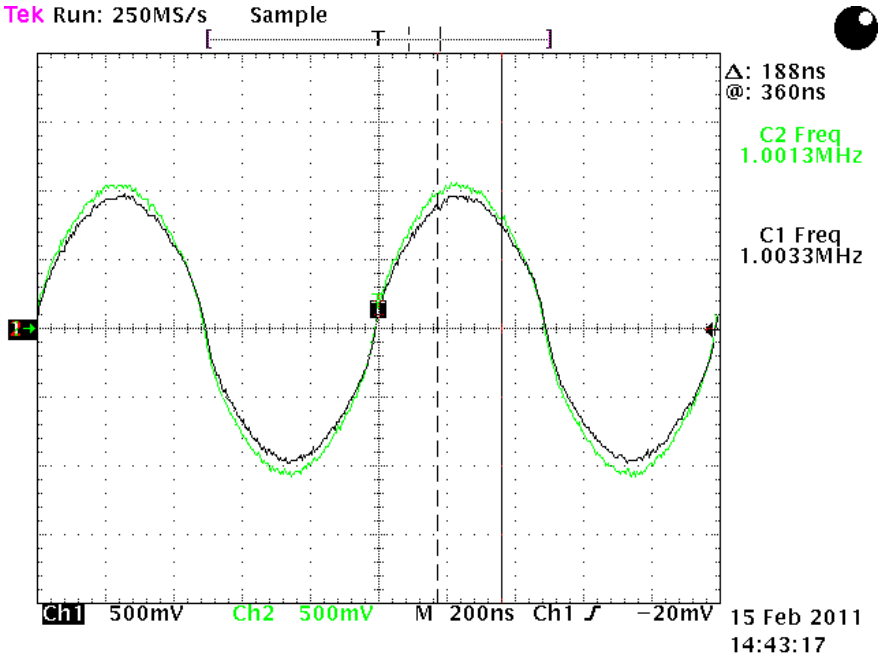


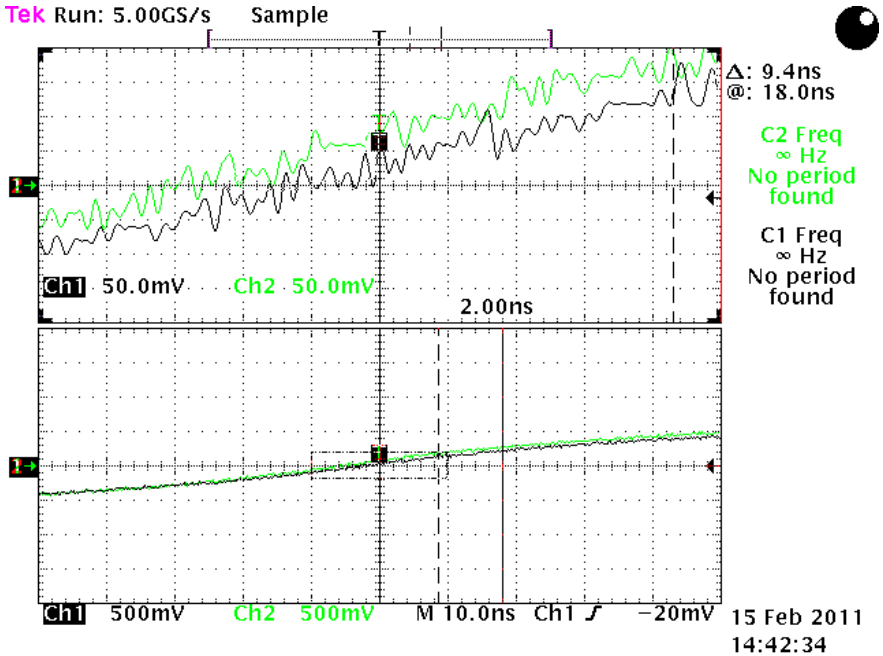
Green: EVM#1

Black: EVM#2

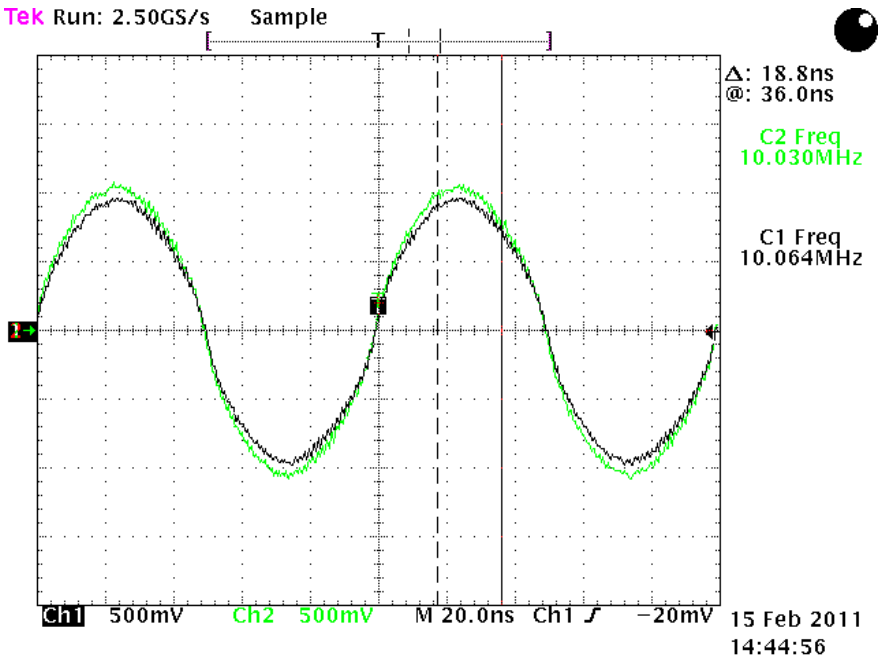
Output skew ~ 100ps

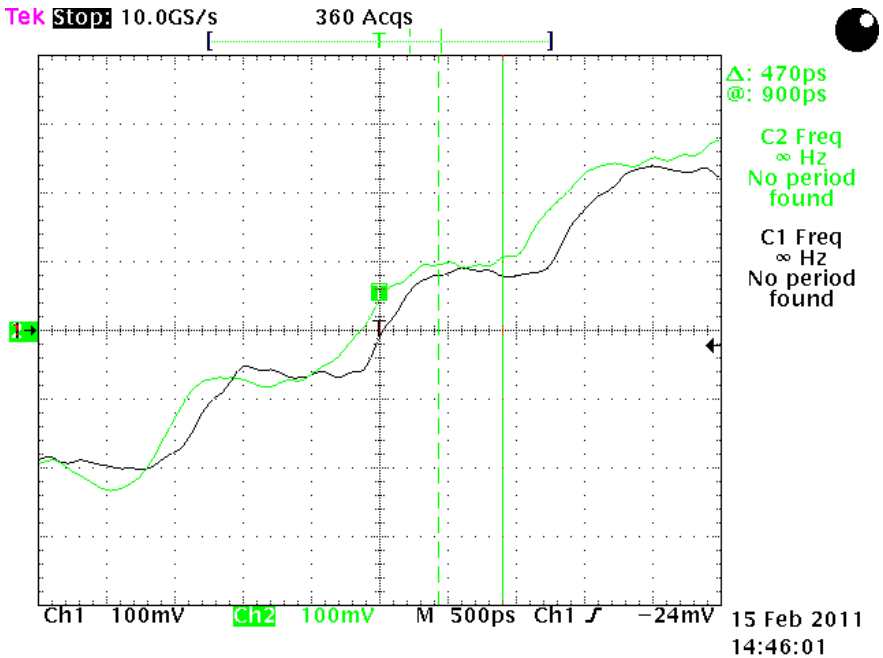
IF=1MHz, Baseband = 1MHz, CMIX off



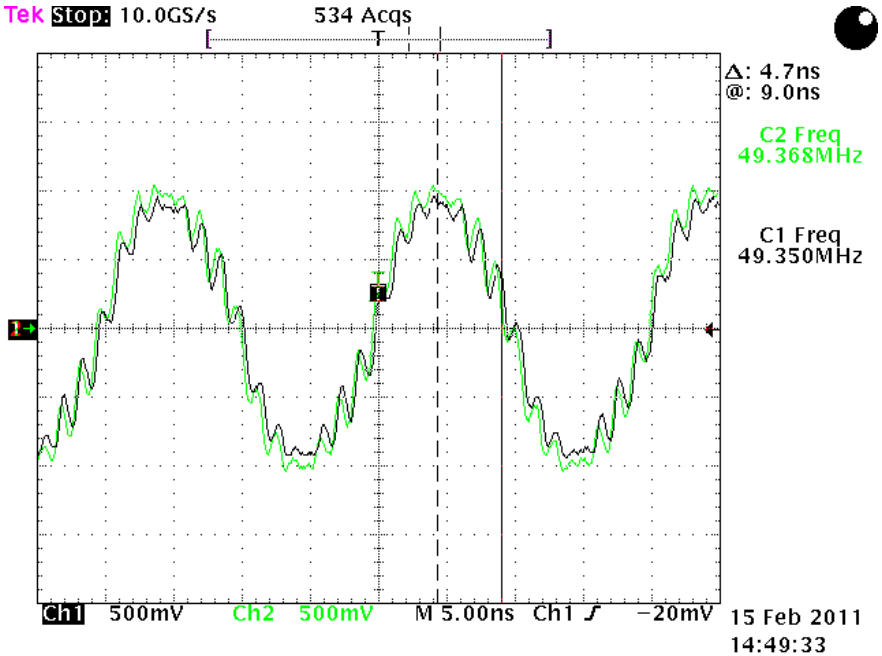


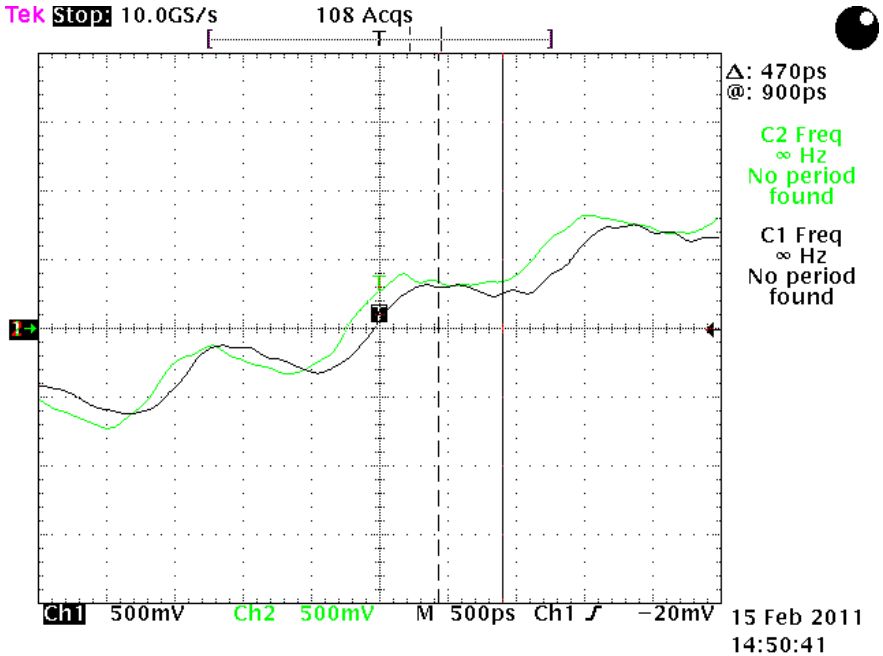
IF=10MHz, Baseband = 10MHz, CMIX off



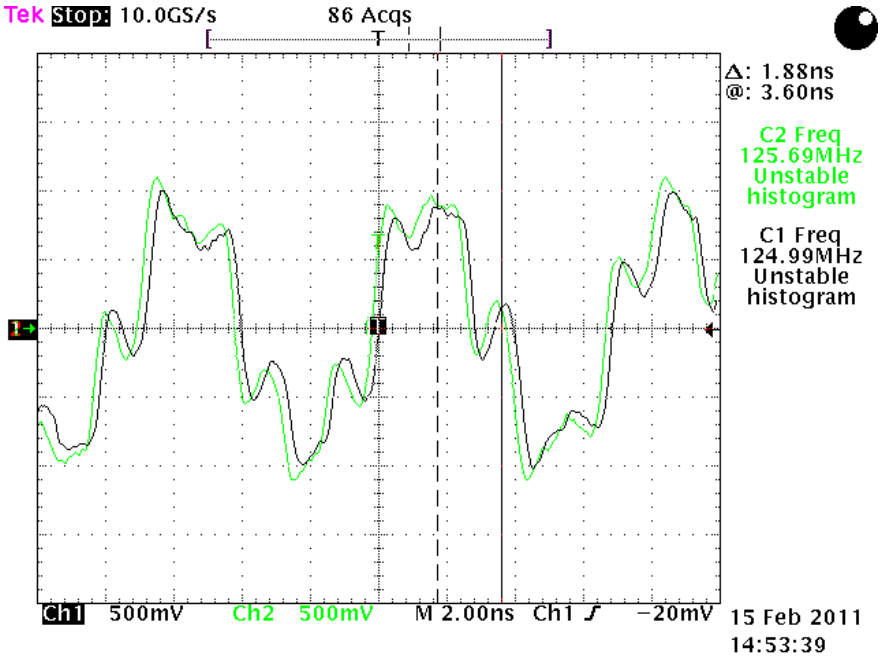


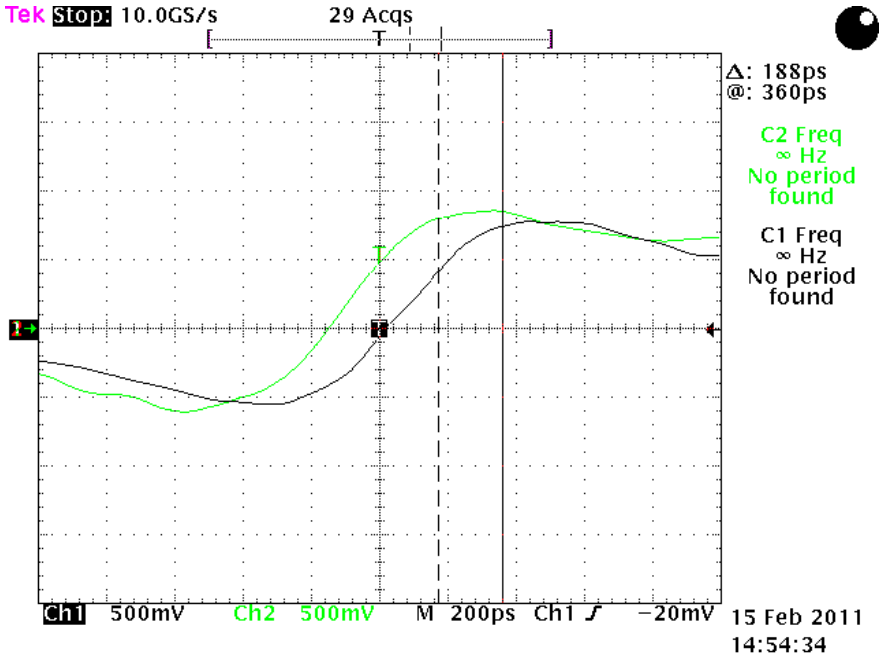
IF=50MHz, Baseband = 50MHz, CMIX off



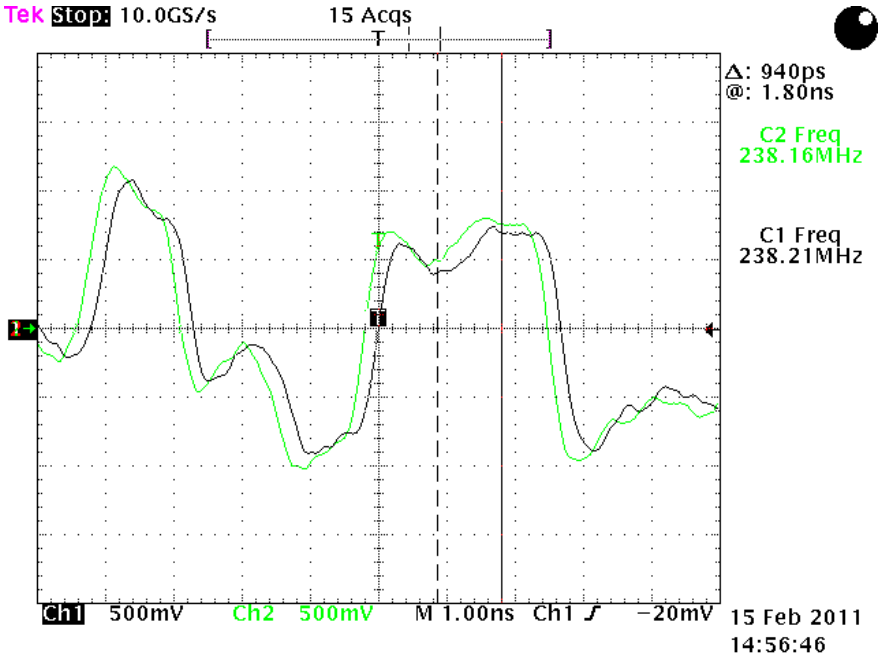


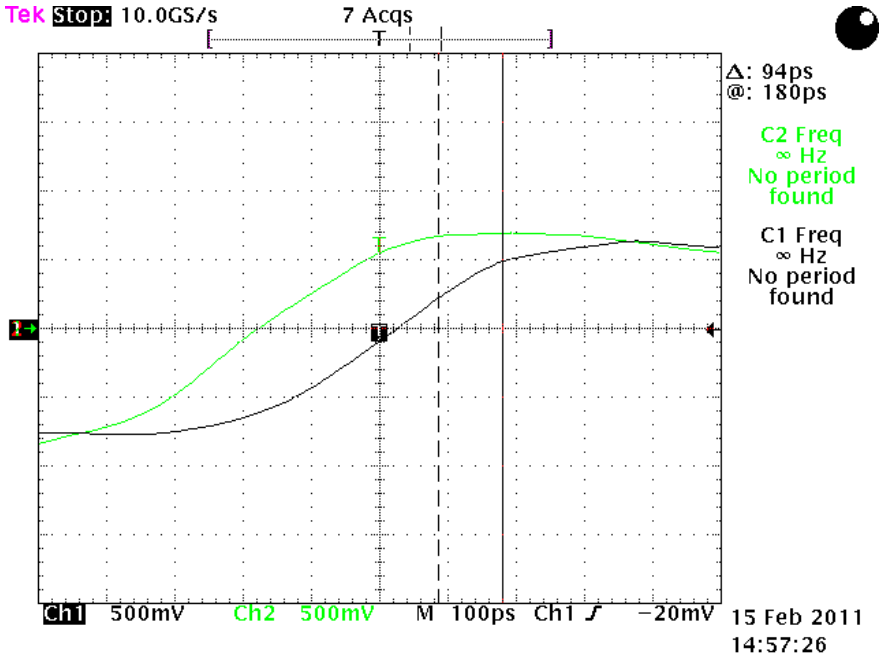
IF=138.32MHz, Baseband = 46.16MHz, Fs/8 CMIX



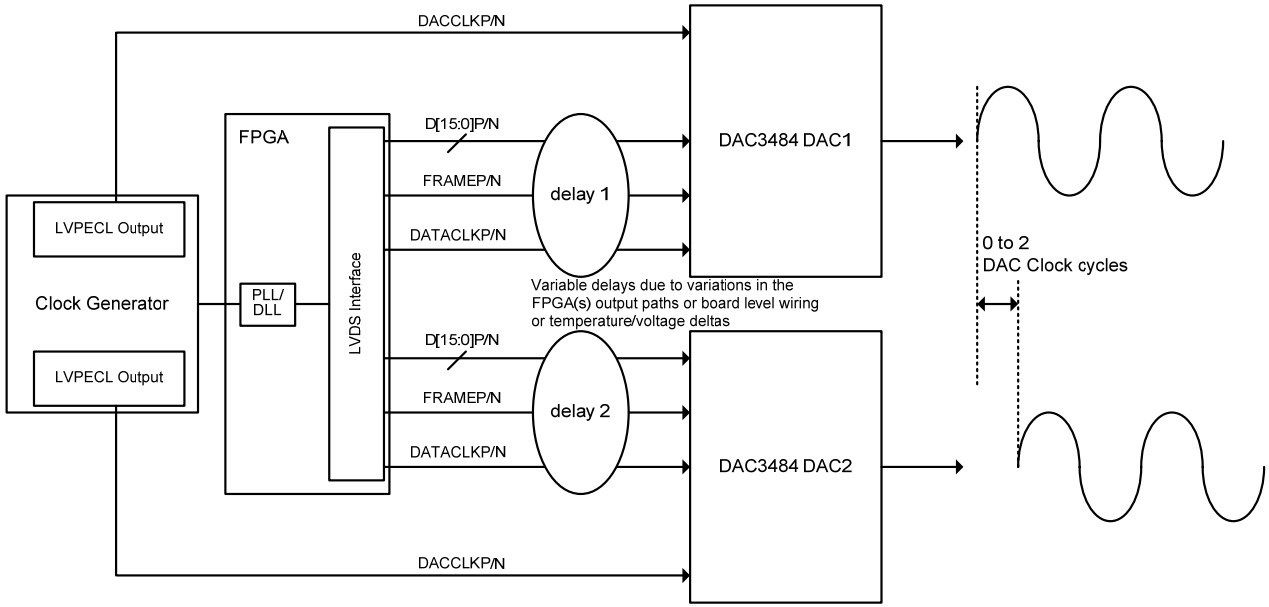


IF=201.6MHz, Baseband = 17.28MHz, Fs/4 CMIX



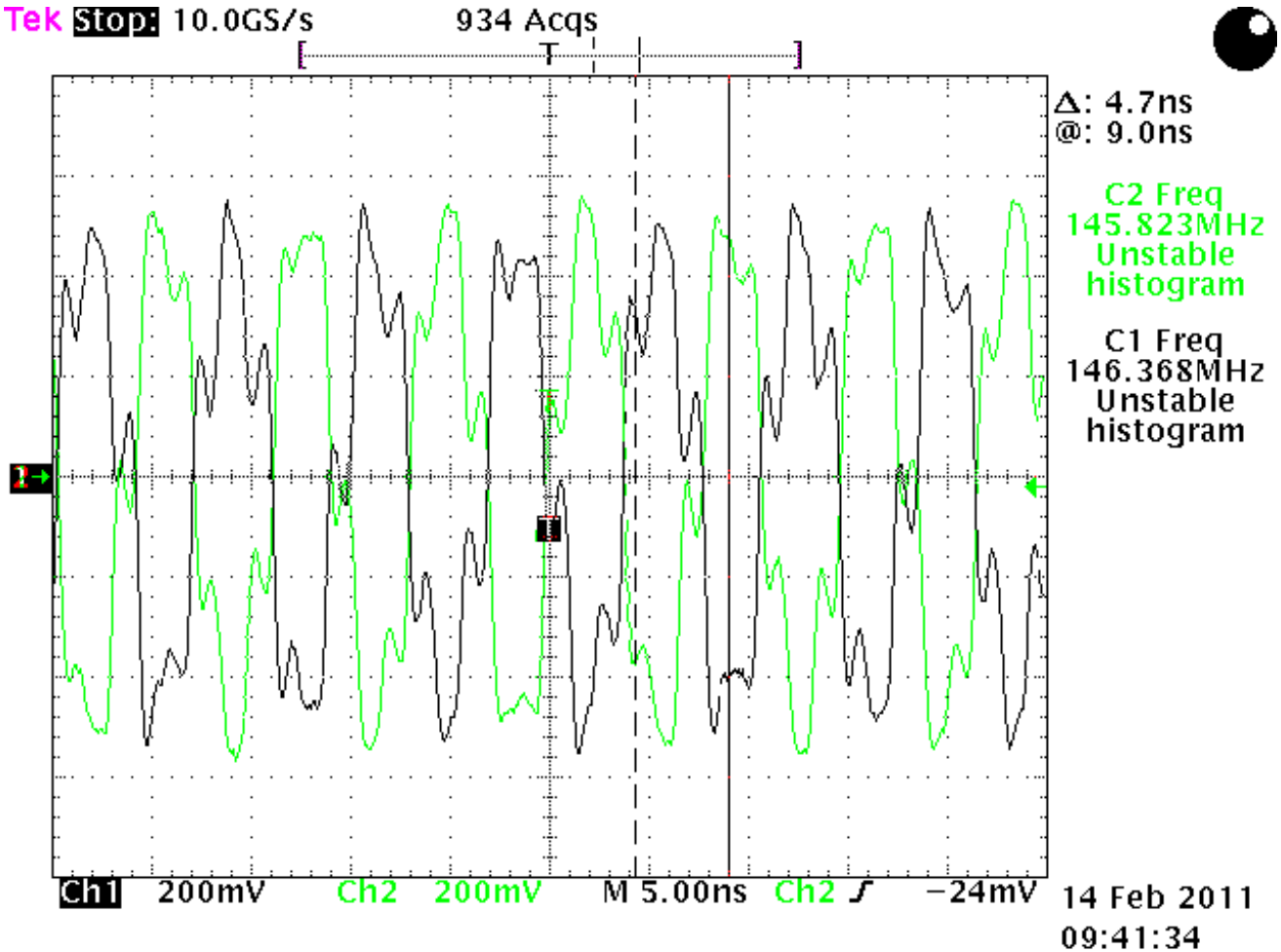


Part 2
Comparison with Single Sync Source Mode



- In Single Sync Source mode, the FIFO write and read pointers are reset from the same sync source, either FRAME or SYNC.
- the FIFO read pointer reset is handoff between the two clock domains (DATACLK and FIFO OUT CLOCK) by simply re-sampling the write pointer reset. Since the two clocks are asynchronous there is a small but distinct possibility of a meta-stablility during the pointer handoff. This meta-stability can cause the outputs of the multiple devices to slip by up to 2 clock cycles.

Single Sync Source Mode IF=140M



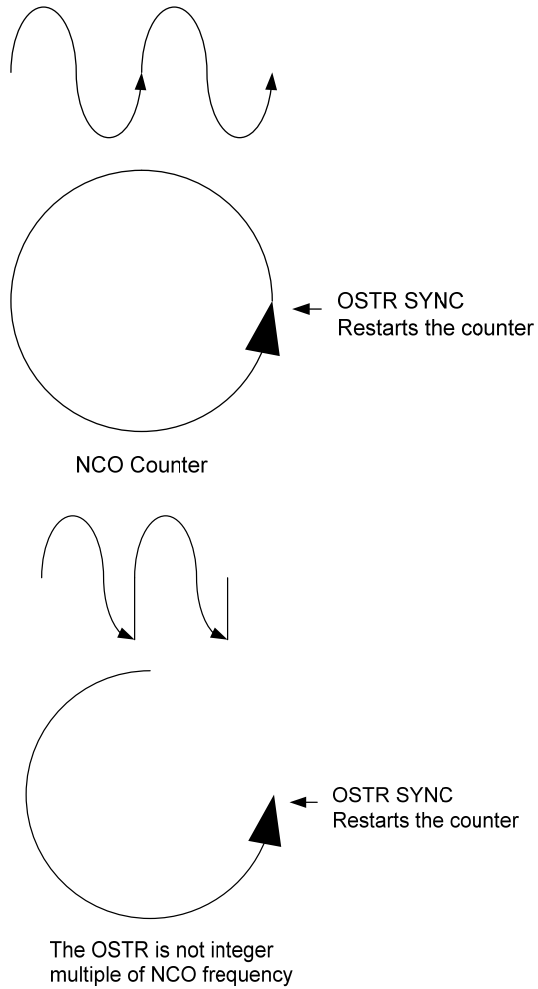
Single Source Mode is unable to achieve the output phase matching due to the potential 0 to 2 DACCLOCK skew between the devices.

Part 3
Synchronization Procedure (NCO synced through OSTR)

1. Set TXENABLE to logic LOW
2. Supply all DAC power supplies
3. Provide all LVPECL inputs: DACCLKP/N and OSTRP/N (special requirement for OSTR frequency).
4. Toggle the RESETB pin for a minimum 25ns active low pulse width
5. Program the SIF Registers to configuration the DAC3484
    1. Config31, bit<3:2> syncsel_dataformmatter = “00” to sync the FIFO data formmatter with LVDS FRAME.
    2. Config32, bit<15:12> syncsel_fifoin = “0010” to sync the FIFO input pointer with LVDS FRAME
    3. Config32, bit<11:8> syncsel_fifoout = “0100” to sync the FIFO output pointer with LVPECL OSTR
    4. Config32, bit<0> clkdiv_sync_sel = “0” to point clock divider sync source to LVPECL OSTR
    5. Config0, bit<2> clkdiv_sync_ena = “1” to enable the clock divider sync to sychronize the phase of the internal divided-down clocks
6. Special Programming Procedure for NCO sync: all the NCO block synced by OSTR signal.
    1. Config31, bit<15:12> = 0010
    2. Config31, bit<11:8> = 0010
    3. Config31, bit<7:4> = 0010
7. Provide all LVDS inputs (D[15:0]P/N, DATACLKP/N, FRAMEP/N, SYNCP/N, and PARITYP/N) simultaneously.
8. The FIFO input/output pointers and clock divider should be synced at this point by rising edge of FRAMEP/N and OSTRP/N.
9. Enable the data transmission by asserting the TXENABLE to logic HIGH or sif_txenable = “1”.

NCO Syncing

- NCO counter will cycling through the sinewave values at the desired NCO frequency.
- A sync signal, such as OSTR sync, will restart the counter so the counter start counting from 0 until the cycle ends and restart.
- If the OSTR signal is a repeating signal, the frequency must equal to the NCO frequency to allow to counter to finish counting the sinewave cycle.
- The NCO frequency can be integer multiples of OSTR frequency to allow multiple cycling of counter with one sync.
- If the periodic OSTR signal does not allow the complete cycling of the NCO counter, the output sinewave will be distorted.



Meeting both FIFO output pointer sync and NCO sync requirement

- **Periodic OSTR Signal**: If both the FIFO output pointer sync and the NCO sync are pointed to the OSTR signal, then the OSTR signal frequency and NCO frequency must meet the requirement for both blocks.
  - OSTR requirement for FIFO output point sync
    - FDAC/(n*interpolation*8), where n is an integer.
    - This allows maximum frequency to re-sync the FIFO every 8 samples.
  - OSTR requirement for NCO sync
    - Fnco/m, where m is an integer.
    - This allows maximum frequency to complete the NCO counter cycle once.
  - To meet both requirements,
    - Fostr = Fnco/(m*n*interpolation*8) given FDAC/Fnco must be an integer
- **Single Pulse OSTR Signal**
  - There is no requirement for NCO frequency to be evenly divisible by FDAC if a single pulse OSTR signal is used to sync the FIFO and NCO during initialization.

Derivation

$$F_{OSTR\_FIFO} = \frac{F_{DAC}}{n \times interpolation \times 8} \text{ given n is an integer}$$

$$F_{OSTR\_NCO\_SYNC} = \frac{F_{NCO}}{m} = \frac{F_{DAC}}{\frac{F_{DAC} \times m}{F_{NCO}}} \text{ given m is an integer}$$

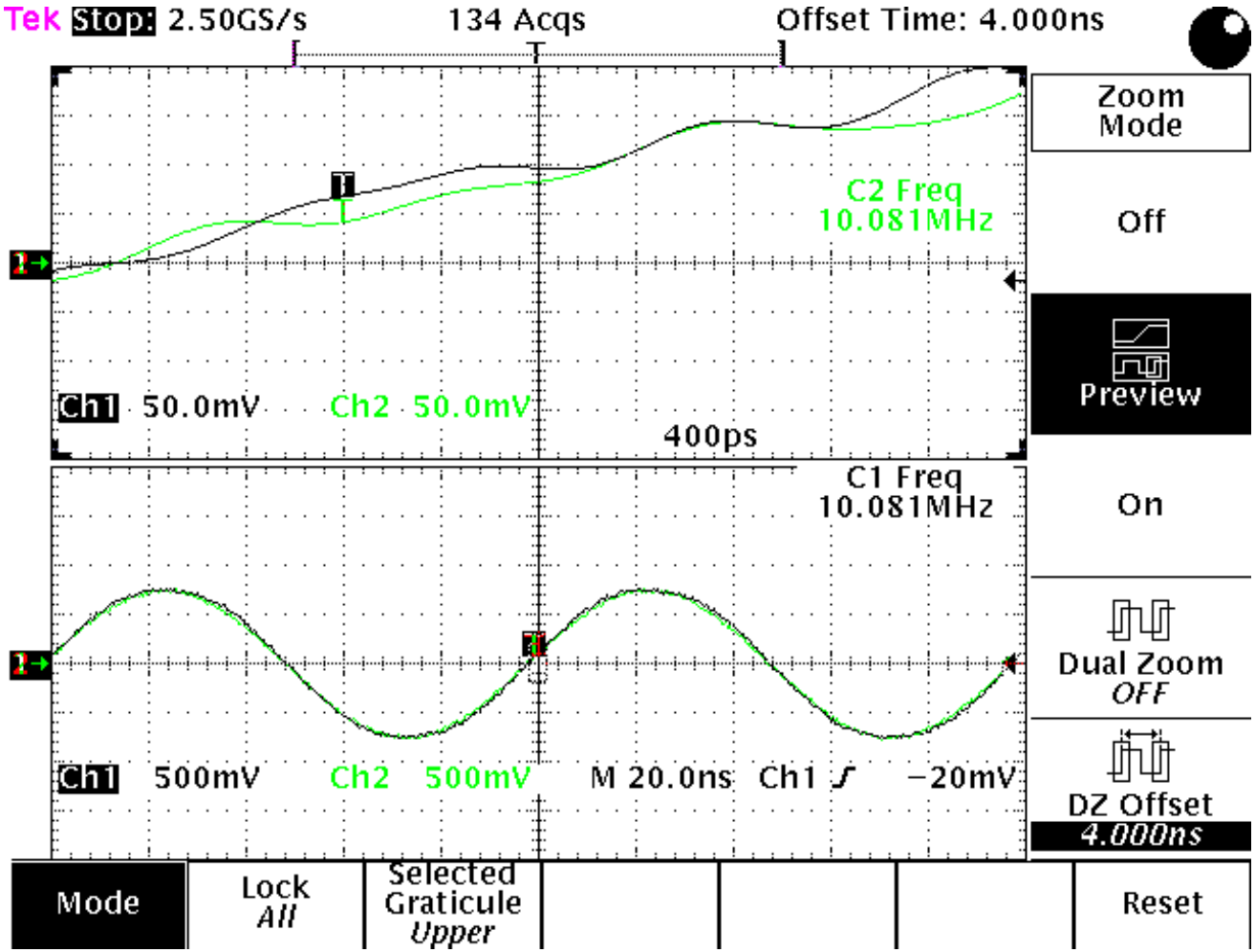
$$F_{OSTR\_FIFO\_NCO\_SYNC} = \frac{F_{DAC}}{LCM(n \times interpolation \times 8, \frac{F_{DAC} \times m}{F_{NCO}})} \text{ given } \frac{m \times F_{DAC}}{F_{NCO}} \text{ is an integer to find LCM.}$$

$$F_{OSTR\_FIFO\_NCO\_SYNC} = \frac{F_{DAC}}{n \times interpolation \times 8 \times \frac{F_{DAC} \times m}{F_{NCO}}}$$

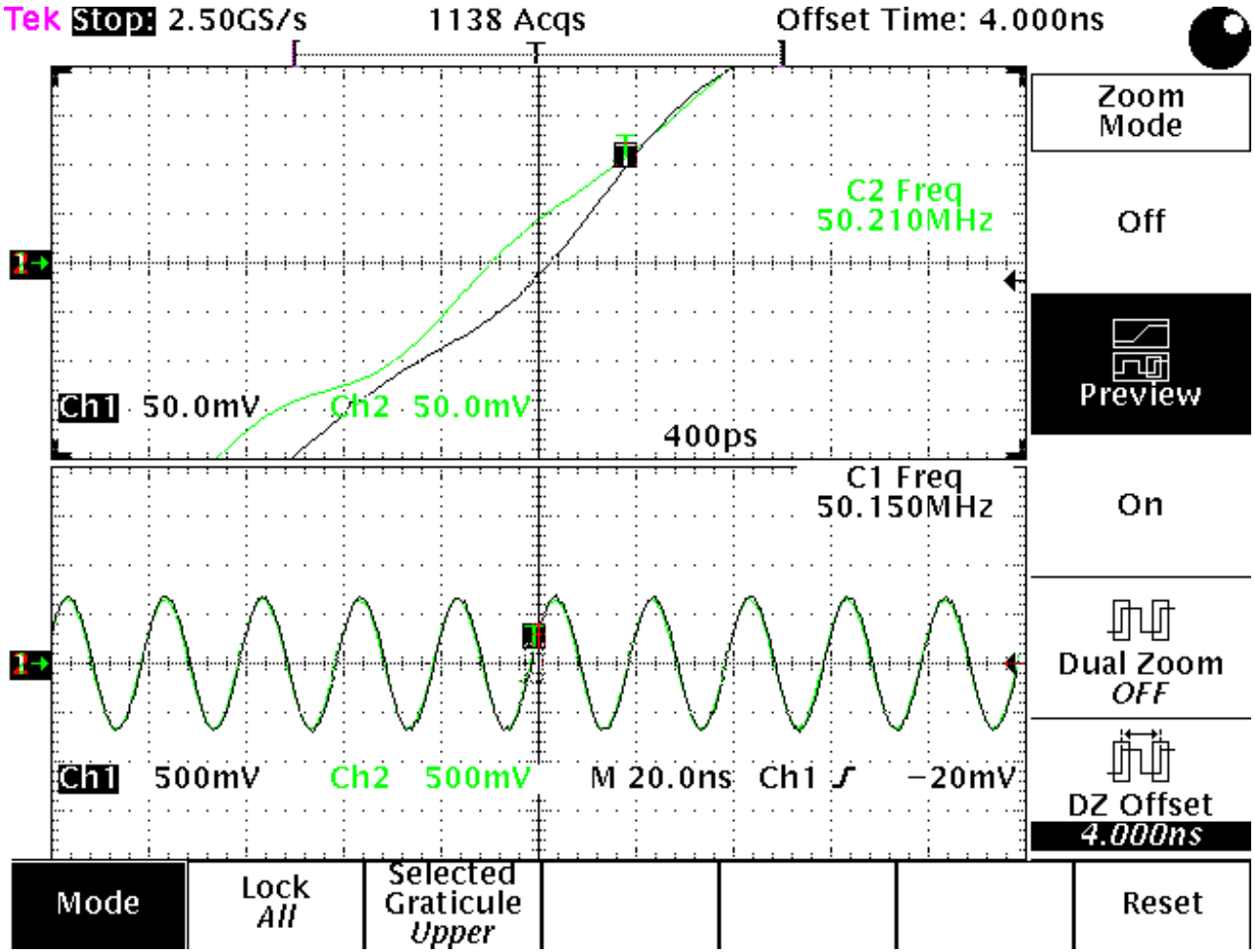
$$= \frac{F_{NCO}}{m \times n \times interpolation \times 8}$$

*LCM = least common multiple.

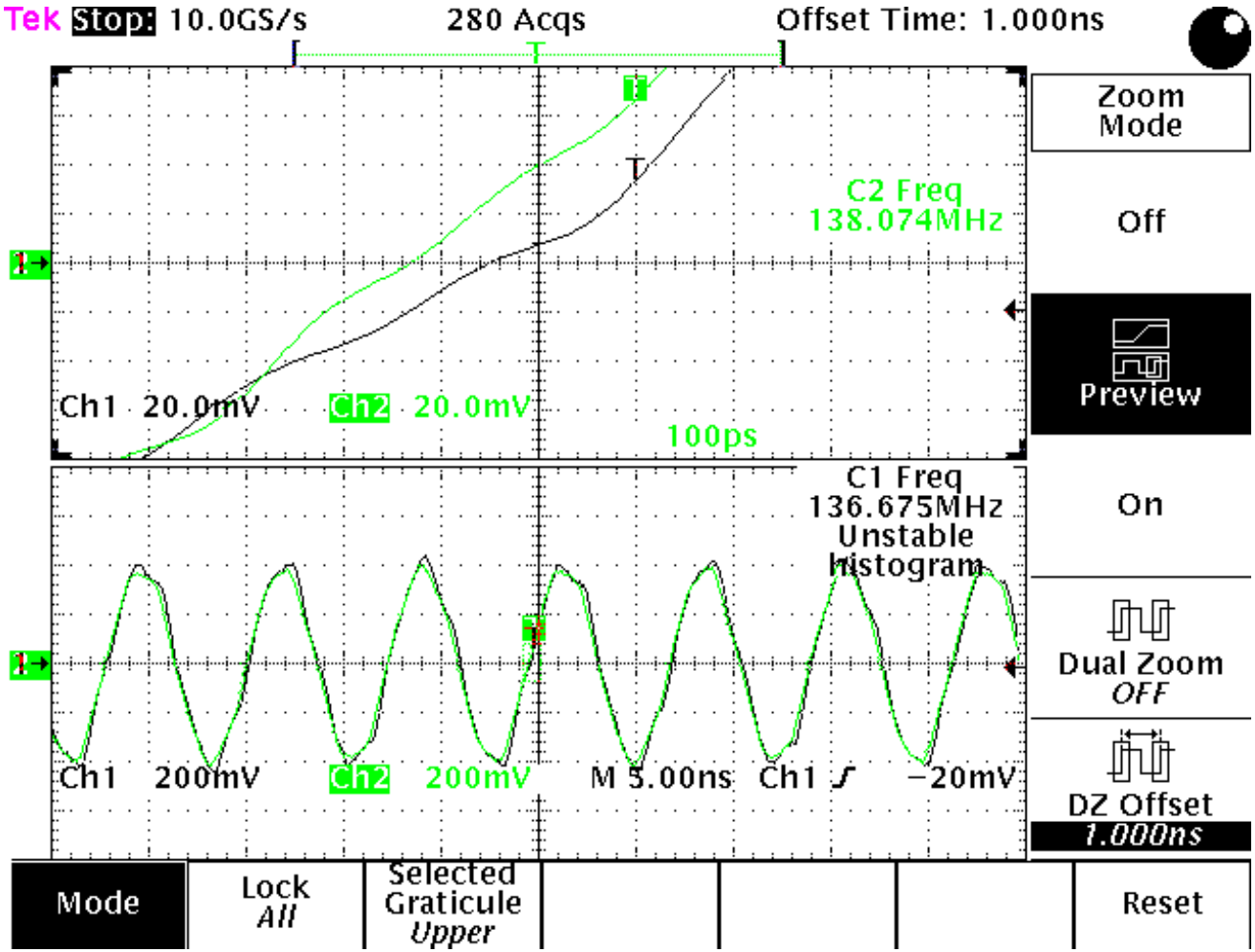
IF = 10MHz, baseband = -13.04MHz, NCO = 23.04MHz (1x OSTR frequency: OSTR signal set to fs/32)



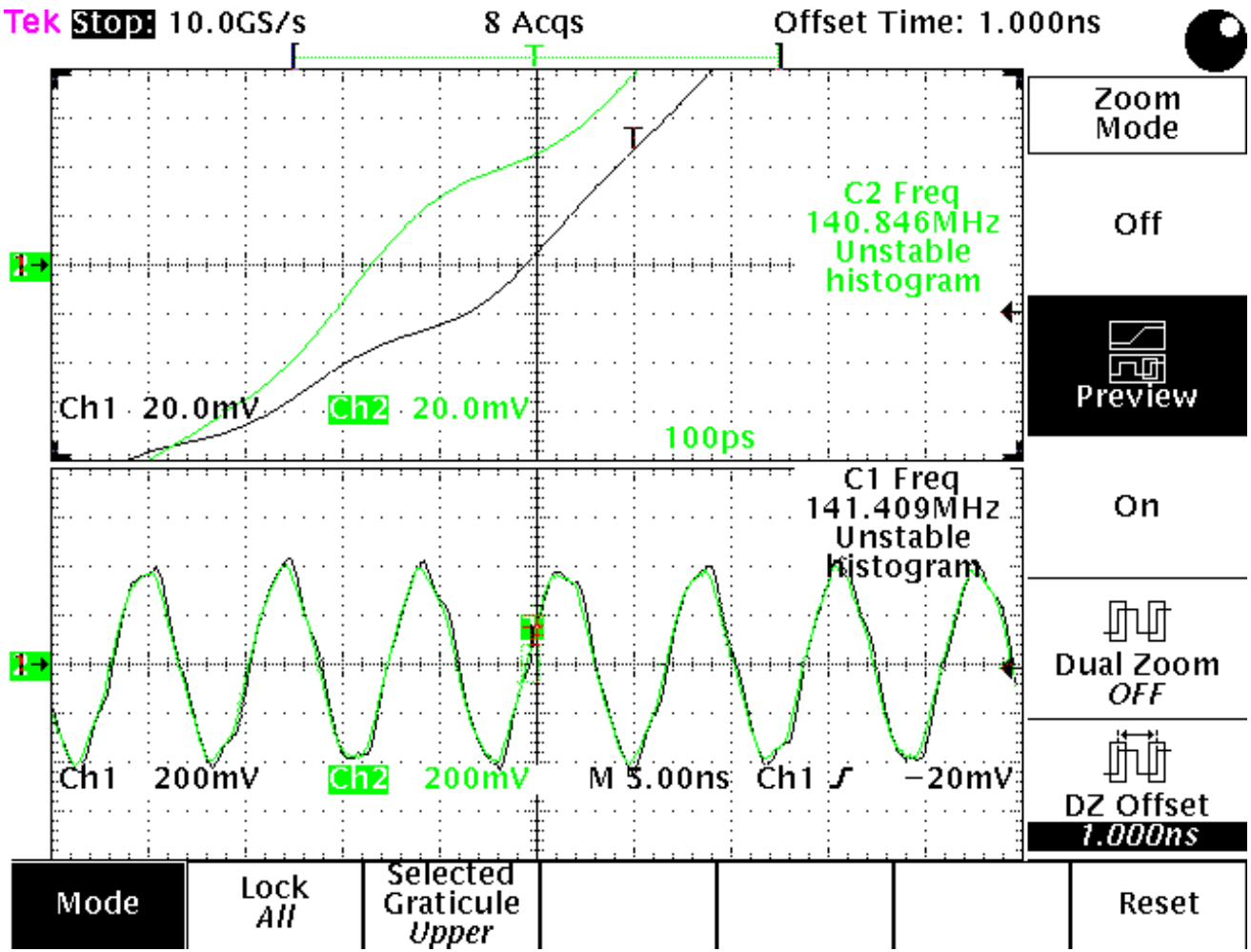
IF = 50MHz, baseband = 3.92MHz, NCO = 46.08MHz (2x the OSTR frequency)



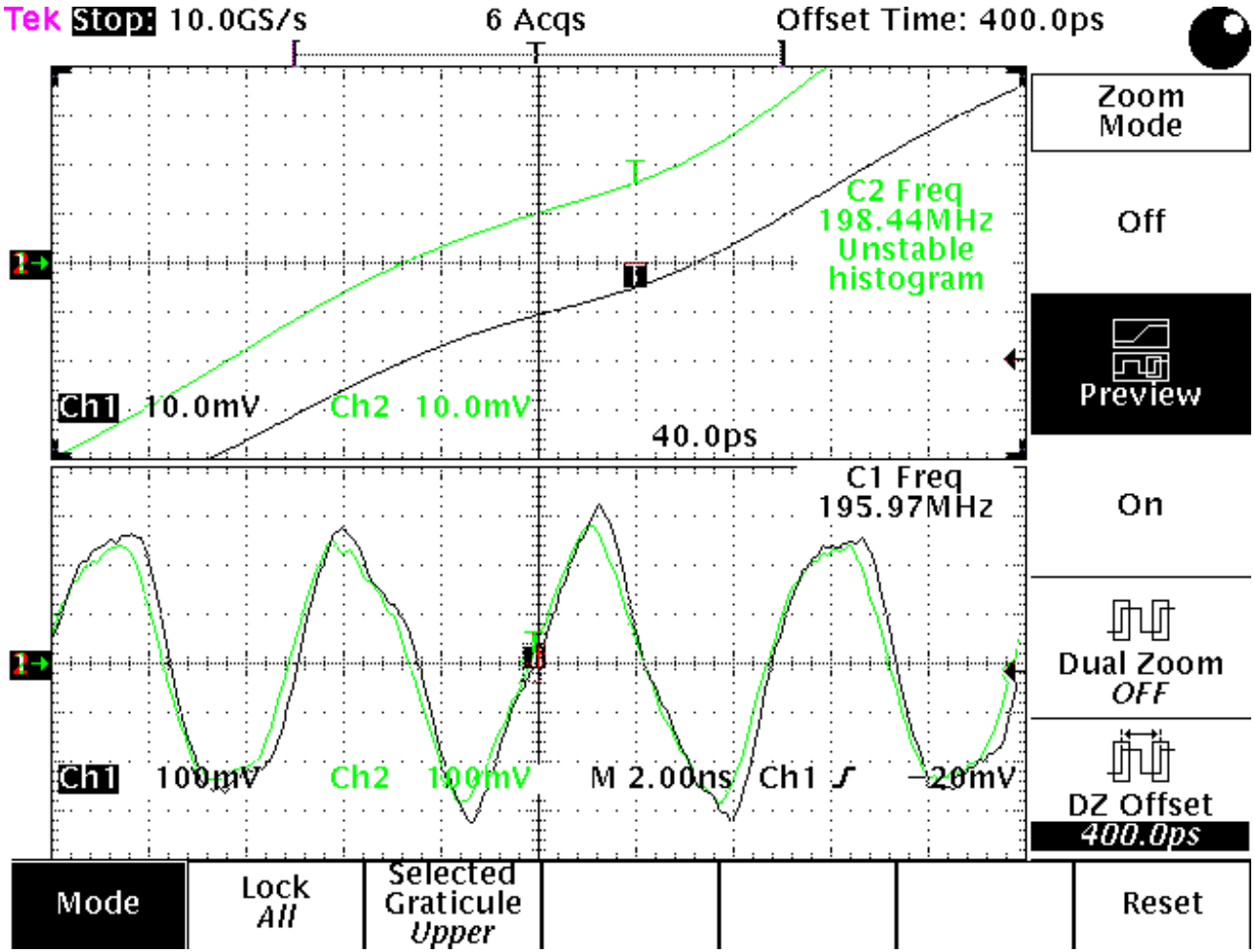
IF = 138.24MHz, baseband = 0MHz,
NCO = 138.24MHz (6x the OSTR frequency)



IF = 140MHz, baseband = 1.76MHz,
NCO = 138.24MHz (6x the OSTR frequency)



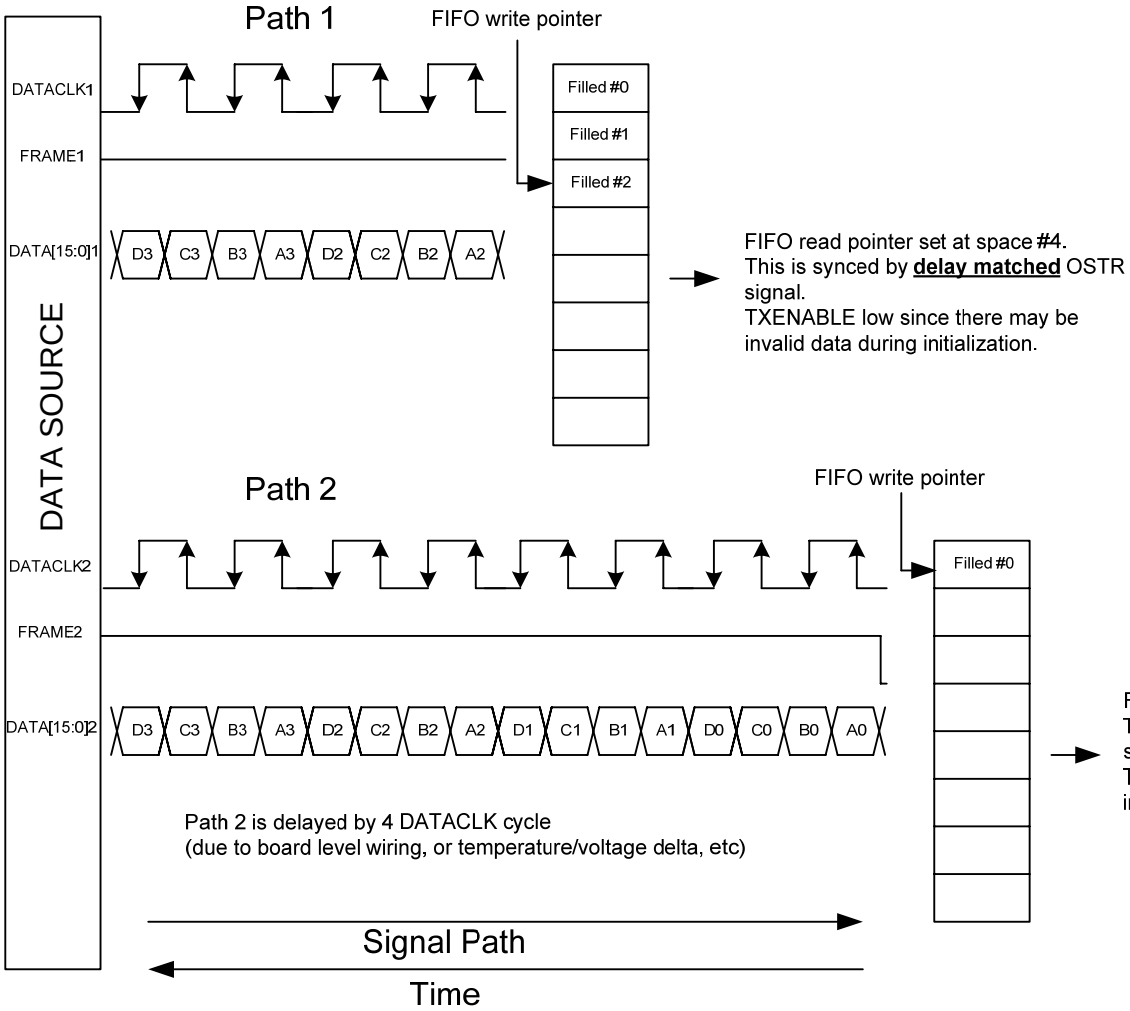
IF = 200MHz, baseband = 15.68MHz,
NCO = 184.32MHz (8x the OSTR frequency)



Part 4
Synchronization Procedure (NCO synced through SYNC or FRAME)

1. Set TXENABLE to logic LOW
2. Supply all DAC power supplies
3. Provide all LVPECL inputs: DACCLKP/N and OSTRP/N (special requirement for OSTR frequency).
4. Toggle the RESETB pin for a minimum 25ns active low pulse width
5. Program the SIF Registers to configuration the DAC3484
    1. Config31, bit<3:2> syncsel_dataformmatter = “00” to sync the FIFO data formmatter with LVDS FRAME.
    2. Config32, bit<15:12> syncsel_fifoin = “0010” to sync the FIFO input pointer with LVDS FRAME
    3. Config32, bit<11:8> syncsel_fifoout = “0100” to sync the FIFO output pointer with LVPECL OSTR
    4. Config32, bit<0> clkdiv_sync_sel = “0” to point clock divider sync source to LVPECL OSTR
    5. Config0, bit<2> clkdiv_sync_ena = “1” to enable the clock divider sync to sychronize the phase of the internal divided-down clocks
6. Provide all LVDS inputs (D[15:0]P/N, DATACLKP/N, FRAMEP/N, SYNCP/N, and PARITYP/N) simultaneously.
7. The FIFO input/output pointers and clock divider should be synced at this point by rising edge of FRAMEP/N and OSTRP/N.
8. At this point, the SYNC signal can be used to synchronize the NCO frequency. Even though the SYNC signal is coming from the data source and is latched in by the DATACLK, the SYNC signal is treated as a data sample. If all the FIFO among the DAC devices are synchronized, all the SYNC signal will be read from the FIFO at the same time, thus the all the NCO is synced.
9. Special Programming Procedure for NCO sync: all the NCO block synced by SYNC signal.
    1. Config31, bit<15:12> = 0100
    2. Config31, bit<11:8> = 0100
    3. Config31, bit<7:4> = 0100
10. SYNC or FRAME signal requirement:
    1. **A single pulse SYNC signal can be used**
    2. **If periodic SYNC signal is used, make sure the frequency can allow the NCO accumulator to complete the count.**
    3. **Like SYNC, the FRAME signal can also be used to synchronize the NCOs. If this is the case, make sure the FIFO input and Data Formatter is not synchronized by the FRAME signal to avoid FIFO error.**
11. Enable the data transmission by asserting the TXENABLE to logic HIGH or sif_txenable = “1”.

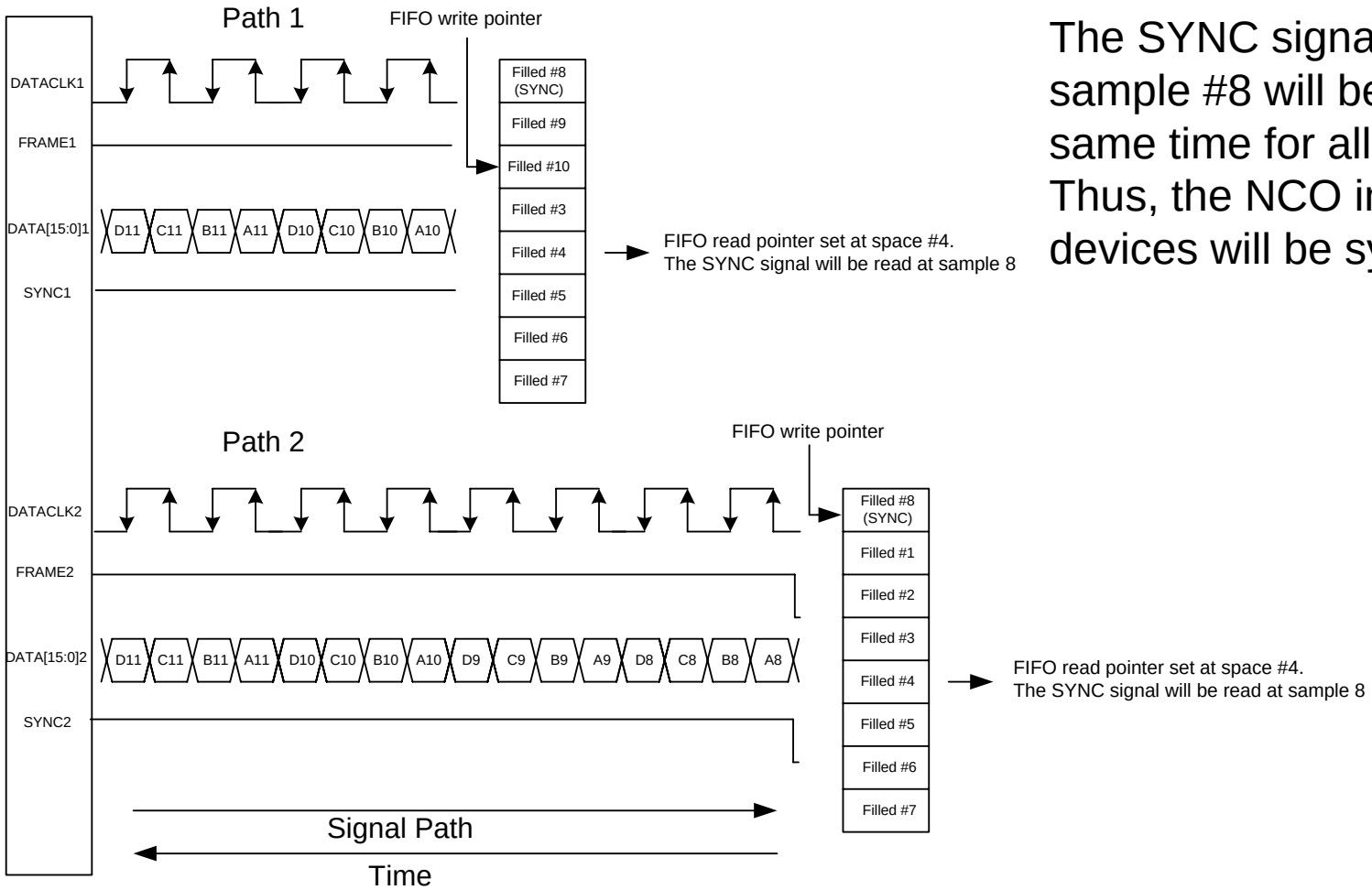
Why can SYNC or FRAME source be used?



FIFO is used to absorb timing variation at the input data side.

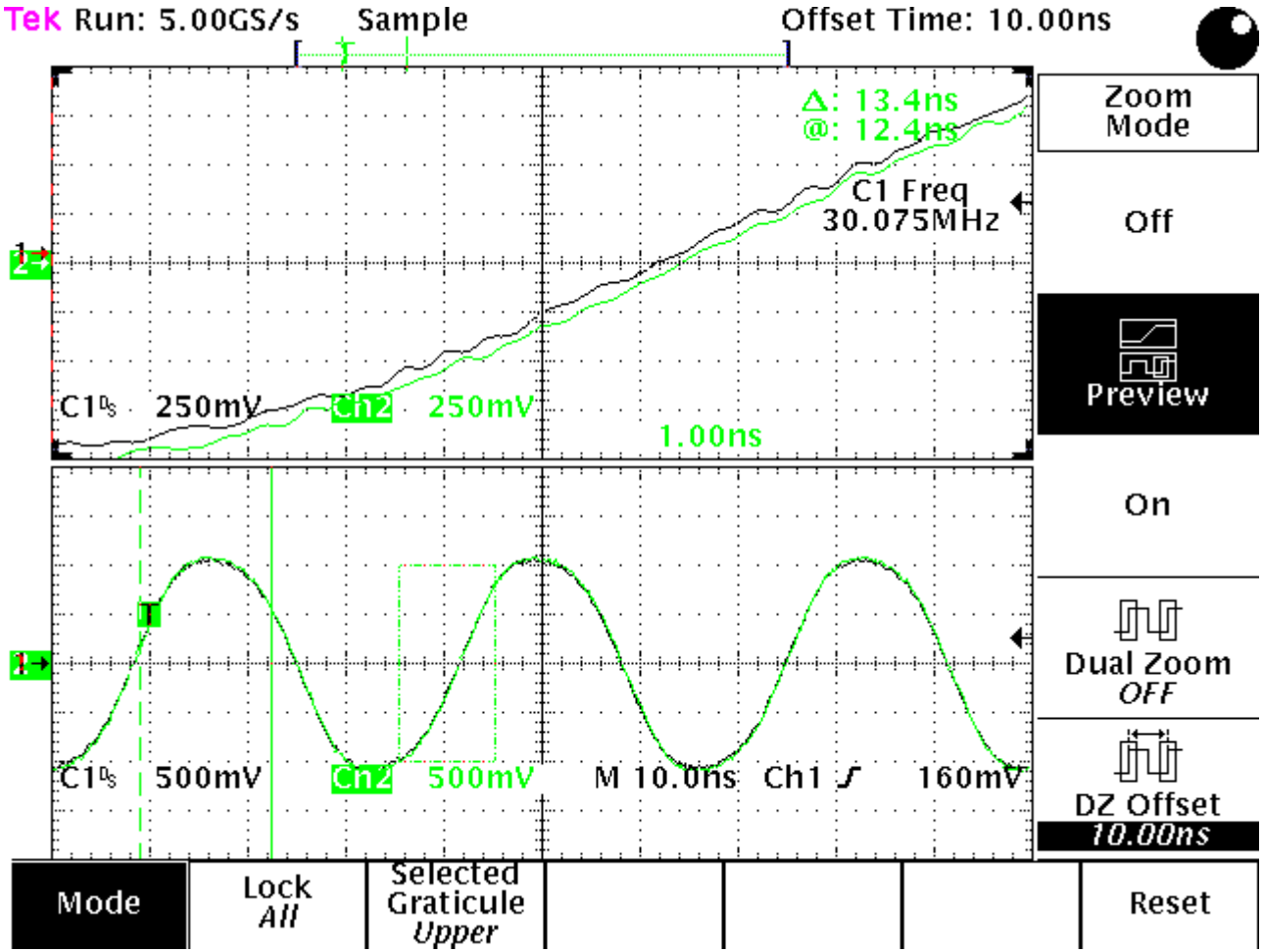
If the SYNC or FRAME signals are send at the same time at the source, then the Dual SYNC Source mode will allow the signals to be read at the same time for all the devices.

Why can SYNC or FRAME source be used?

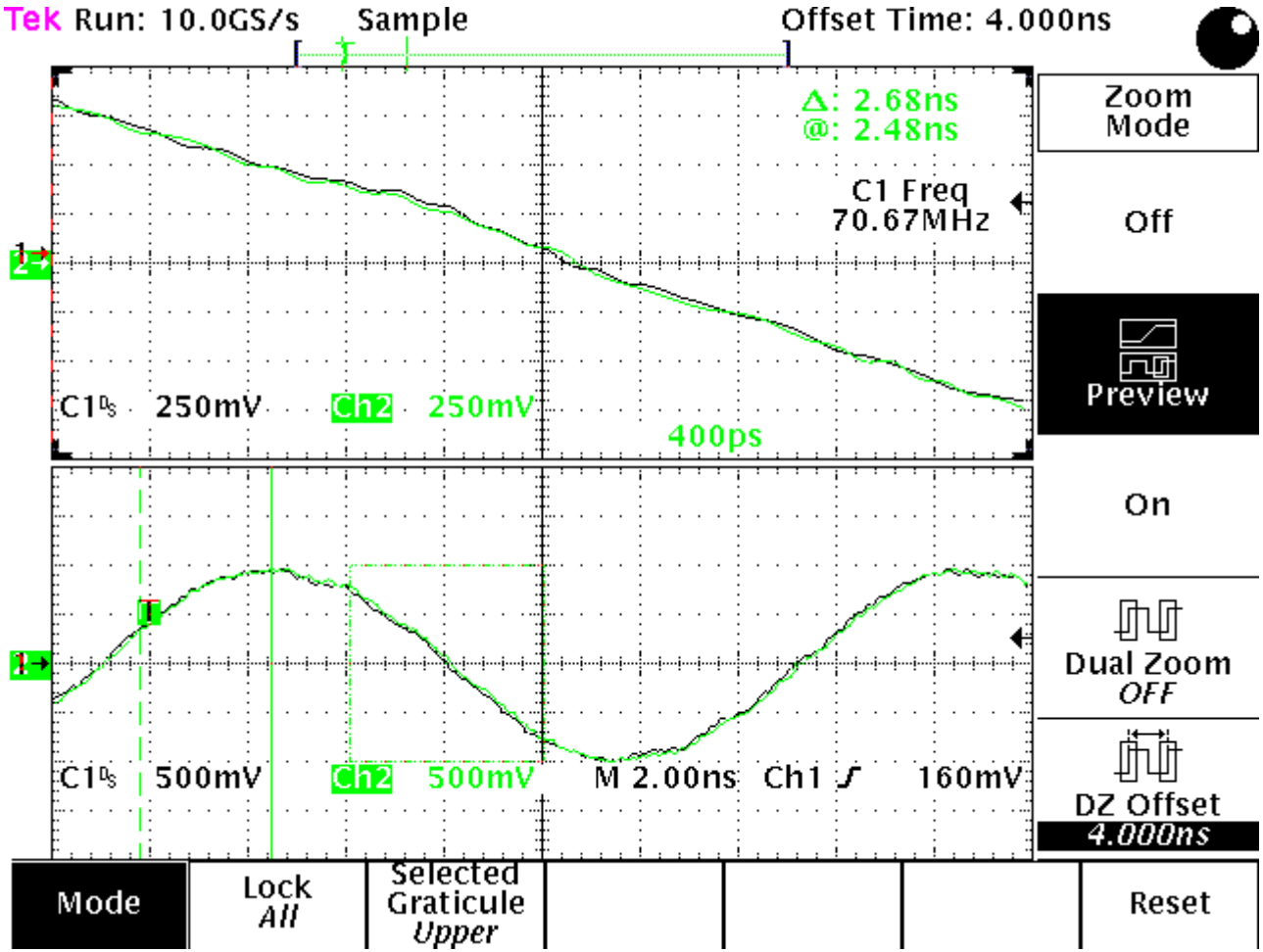


The SYNC signal send around sample #8 will be read at the same time for all the devices. Thus, the NCO in multiple devices will be synced.

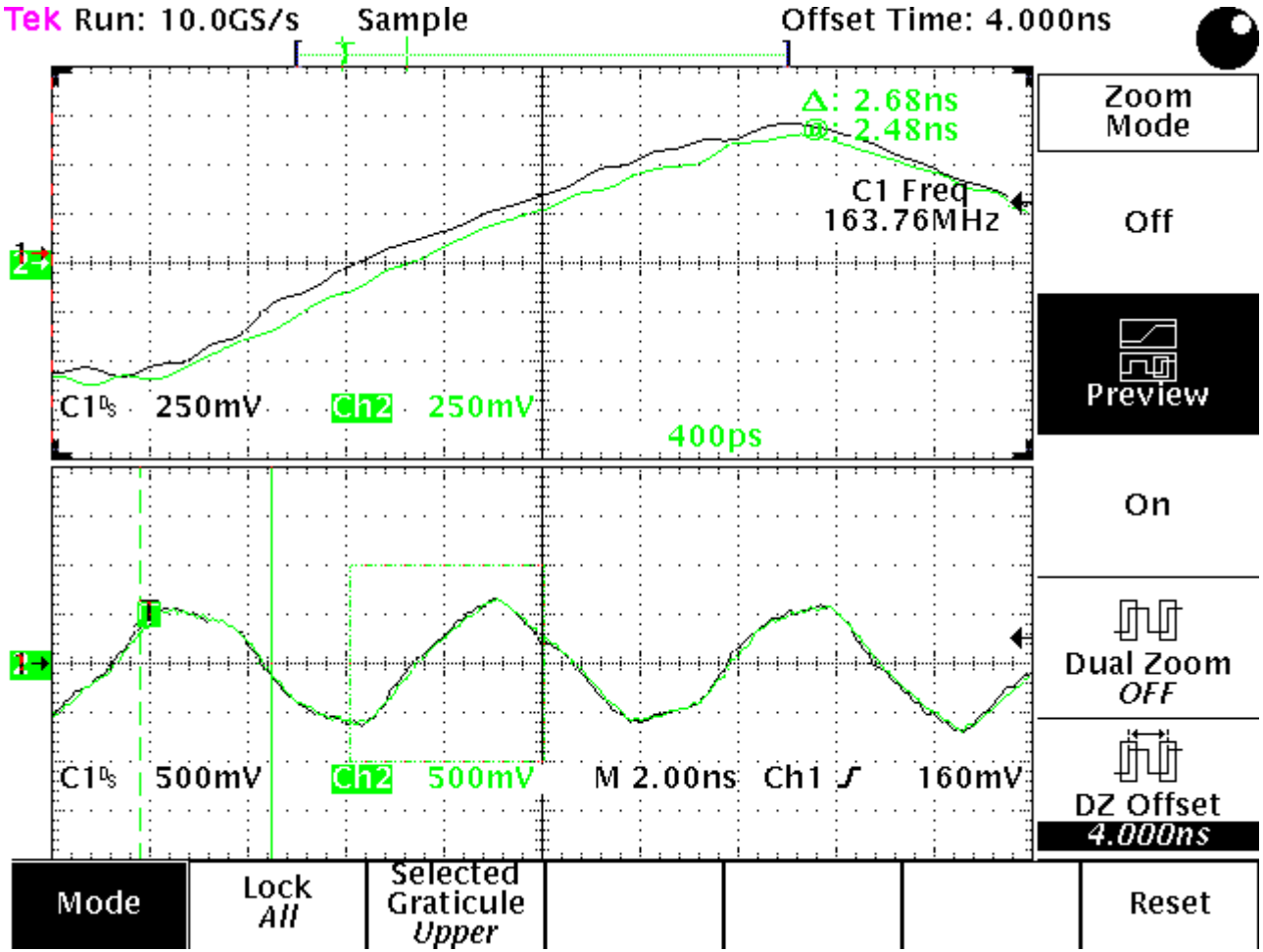
IF = 20MHz, NCO = 10MHz



IF = 20MHz, NCO = 50MHz



IF = 20MHz, NCO = 138.32MHz



IF = 20MHz, NCO = 201.6MHz