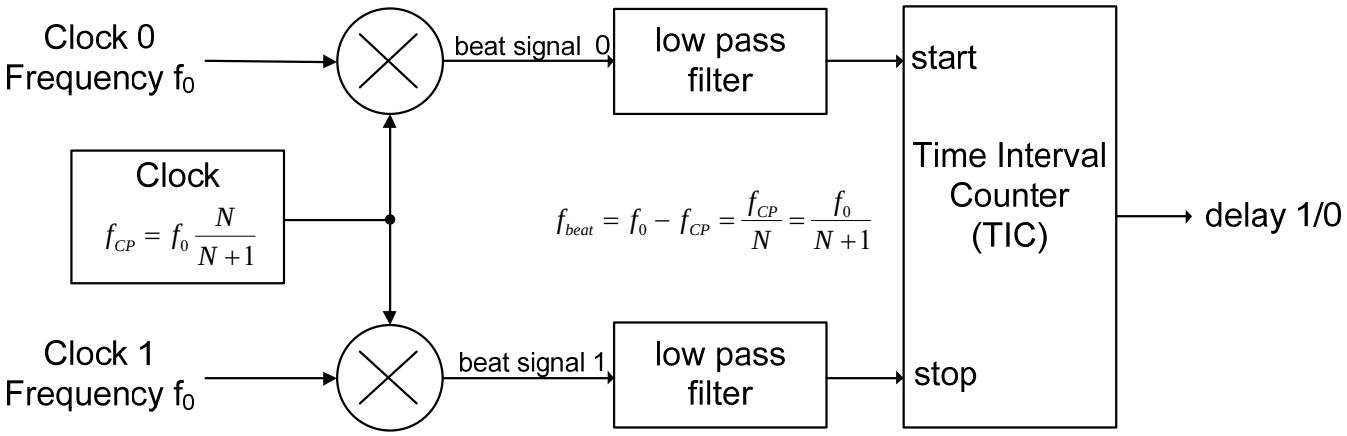
Arithmetic-DMTD

Glitch-free femtosecond time difference measurements for digital clocks

1. Introduction

The digital design described herein is a generalization of the Digital Dual Mixer Time Difference used in CERN's White Rabbit system and described in [1]. The goal here is to overcome the DDMTD glitch filter design problem by suppressing it (no glitch generation), while keeping low FPGA resources in a quite straightforward implementation.

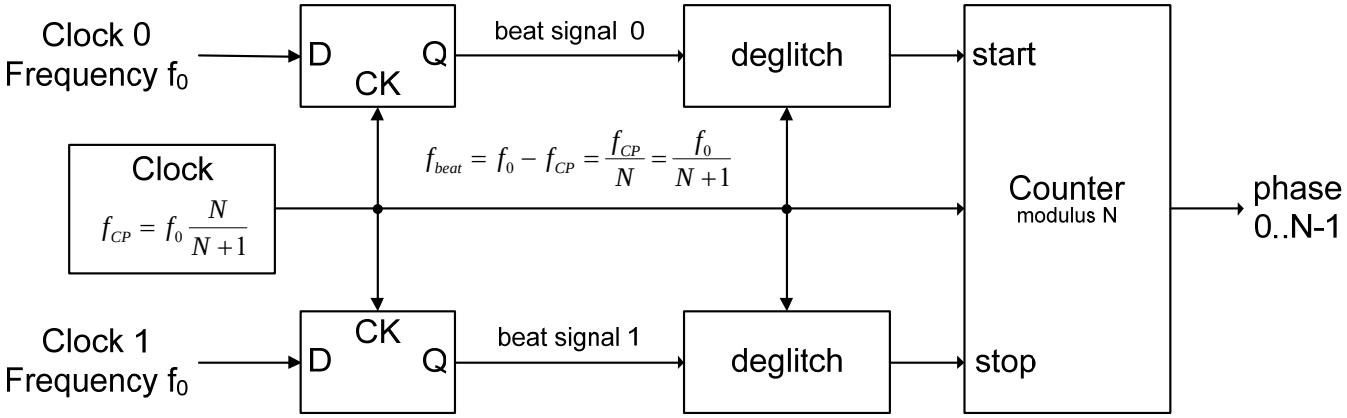
The original (analog) DMTD [2] uses the differential phase conservation property of two clock signals of same frequency f_0 after mixing with a third one of frequency f_{CP}. Letting $f_{CP} = f_0 - f_{beat}$, two signals at beat frequency f_{beat} are produced after mixing and low pass filtering. By choosing f_{beat} low, the phase difference between the beat signals can be measured with high precision by a conventional counter. As such, DMTD is a method to measure precisely the phase between two high frequency signals.



DMTD principle

Since mixers and low pass filters are analog devices, DMTD is difficult to integrate in small designs, especially if numerous measurement channels are required. So, there is a need for a digital counterpart easily fittable into FPGAs.

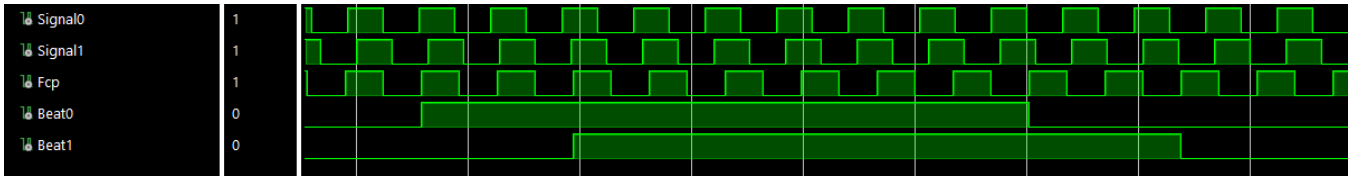
The Digital DMTD (DDMTD) method presented in [1] uses a helper clock of frequency $f_{CP} = f_0 \frac{N}{N+1}$ to sample the input clock signals. It produces two beat signals at frequency $\frac{f_{CP}}{N}$, and the phase is reflected by the reading m of a counter started with one signal and stopped with the second.



DDMTD principle

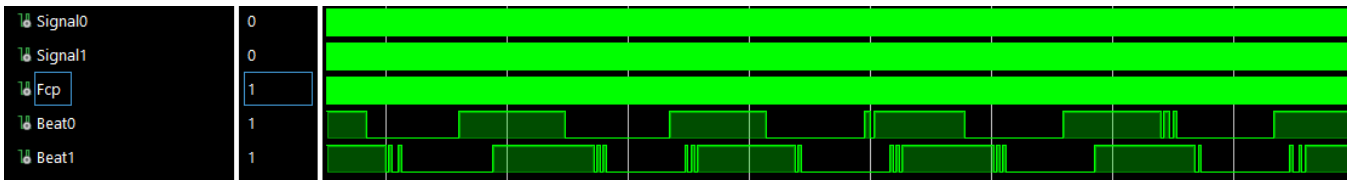
Without jitter, the measure uncertainty is ± 1. Letting the normalized phase in 0..1 instead of 0..2π, the normalized phase measure is then m/N and the normalized phase uncertainty is $\pm 1/N$.

The operation can also be seen like a slow scan of the input signals, with a T_0/N period increment ($T_0 = 1/f_0$).



DDMTD : signals example for N=8

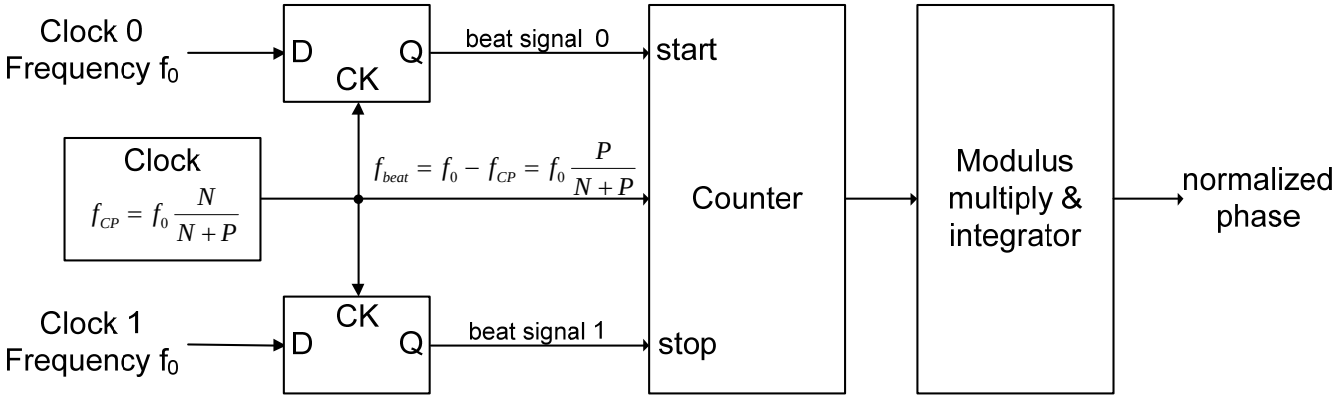
A problem arises when the scan reaches a rising or falling edge : with high N and/or high jitter on signals, the sampling generates random 0s and 1s (glitches) until the edge is scanned. This leads to a fuzzy edge position, and some filtering method is required to estimate the « real » position of the edge. The DDMTD paper [1] discusses several « deglitcher » algorithms.



DDMTD : jitter effect example for high N (here N=32)

2. The ADMTD

One can avoid glitches with a sampling step greater than the signals jitter zone width. This can be achieved simply by using a helper frequency $f_{CP} = f_0 \frac{N}{N+P}$, with $P << N$, and P, N relatively prime. The sampling period is then $T_{CP} = \frac{1}{f_{CP}} = T_0 \frac{N+P}{N} = T_0 \left(1 + \frac{P}{N}\right)$.

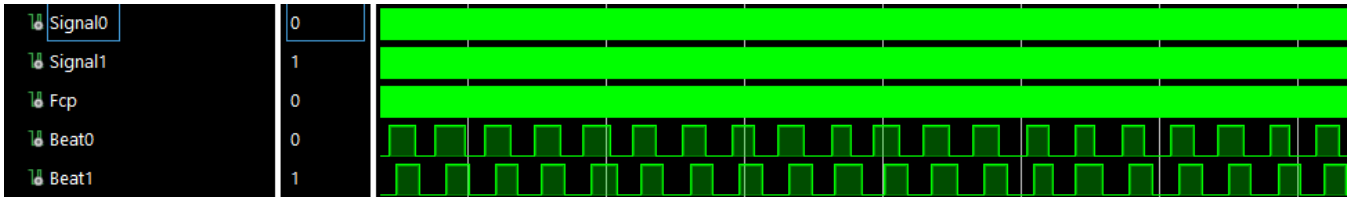


ADMTD : principle

The beat frequency is $f_{beat} = f_0 - f_{CP} = f_0\left(1 - \frac{N}{N+P}\right) = f_0 \frac{P}{N+P}$.

Again, the sampling can be seen as a slow scan of the input signals, with a $P\frac{T_0}{N}$ period increment. As P and N relatively prime, the input signals are still scanned through N differents points, but after P subscans. So the measure resolution is the same than for DDMTD (ie $\frac{T_0}{N}$), but there is no two successive points in the dark edge zone, avoiding glitch production, when P/N is large enough.

By processing the DDMTD example (N=32) with P=5, the glitches disappear :



ADMTD : example of glitch avoidance on the beat signal

Without jitter, each basic measure between signals edges gives a gross value m_i in $[0..T_{beat}/T_{CP}]$, ie a count value in $\{0..ceil(N/P)\}$. The normalized count (in $\{0..N-1\}$) associated with this measure is : ${}^N m_i = \left(\frac{m_i T_{CP}}{T_{beat}} N\right) \bmod N = (m_i P) \bmod N$. The choice $N = 2^n$ eases the implementation of the modulus in a FPGA. Since there now a multiply operation, this method is called here ADMTD (Arithmetic DMTD).

Now, we can average M successive basic measures to get the final differential phase value. M must be multiple of P to insure a uniform scan coverage.
The differential normalized phase can be written $\varphi_N = average\left({}^N m_i\right)$. The average computation is done as : $\varphi_N = \frac{{}^N m_{i_0}}{N} + \frac{1}{M}\sum_{i=i_0}^{i_0+M-1} \frac{{}^N m_i - {}^N m_{i_0}}{N} = \frac{1}{N}\left({}^N m_{i_0} + \frac{1}{M}\sum_{i=i_0}^{i_0+M-1} {}^N m_i - {}^N m_{i_0}\right)$, with ${}^N m_{i_0}$ is a centering value derived from the first basic measure (basic phase measures are supposed not noisy enough to cover more than one quadrant).

The period of the beat signal is : $T_{beat} = \frac{1}{f_{beat}} = \frac{N}{P} T_{CP} = floor\left(\frac{N}{P}\right) T_{CP} + \frac{N \bmod P}{P} T_{CP}$. Without jitter, the edge position errors are $e_p = \left[e_0 + \frac{\left(p\left(N \bmod P\right)\right) \bmod P}{P} T_{CP} \right] \bmod T_{CP}$, $p = 0..P-1$, where $e_0 \in \left[0..\mathrm{T_{CP}}\right[$ is first edge position (same as the DDMTD single-edge start position error).

The errors on the P basic measures (between the edges of the two beat signals) are :

$$\delta_p = \left[e_0 + \frac{\left(p\left(N \bmod P\right)\right) \bmod P}{P} T_{CP} \right] \bmod T_{CP} - \left[e_1 + \frac{\left(p\left(N \bmod P\right)\right) \bmod P}{P} T_{CP} \right] \bmod T_{CP}$$

where $e_1 \in \left[0..\mathrm{T_{CP}}\right[$ is the first (stop) edge error of the second signal.

Since P and N are relatively prime, so are $N \bmod P$ and P. There are also P successive errors values (the sampled sequence of the beat signals repeats every P cycles). One can list the P different errors :

$$\left\{ \delta_k = \left[e_0 + \frac{k}{P} T_{CP} \right] \bmod T_{CP} - \left[e_1 + \frac{k}{P} T_{CP} \right] \bmod T_{CP}, \quad k = 0..P-1 \right\}$$

The averaging of the P basic measures produce a global measure error :

$$\delta = \frac{1}{P} \sum_{k=0}^{P-1} \delta_k = \frac{1}{P} \sum_{k=0}^{P-1} \left[\left[e_0 + \frac{k}{P} T_{CP} \right] \bmod T_{CP} - \left[e_1 + \frac{k}{P} T_{CP} \right] \bmod T_{CP} \right]$$

With $e_0 = \varepsilon_0 + \frac{k_0}{P} T_{CP}$, $0 \le k_0 < P$, $\varepsilon_0 \in \left[0..\frac{T_{CP}}{P} \right[$

and $e_1 = \varepsilon_1 + \frac{k_1}{P} T_{CP}$, $0 \le k_1 < P$, $\varepsilon_1 \in \left[0..\frac{T_{CP}}{P} \right[$, we get :

$$\delta = \frac{1}{P} \sum_{k=0}^{P-1} \left[\left[\varepsilon_0 + \frac{\mathrm{k} + k_0}{P} T_{CP} \right] \bmod T_{CP} - \left[\varepsilon_1 + \frac{\mathrm{k} + k_1}{P} T_{CP} \right] \bmod T_{CP} \right]$$

$$\delta = \frac{1}{P} \sum_{k=0}^{P-1} \left[\varepsilon_0 + \left[\frac{k + k_0}{P} T_{CP} \right] \bmod T_{CP} - \varepsilon_1 - \left[\frac{k + k_1}{P} T_{CP} \right] \bmod T_{CP} \right]$$

$$\delta = \varepsilon_0 - \varepsilon_1 + \frac{1}{P} \sum_{k=0}^{P-1} \left[\left[\frac{k + k_0}{P} T_{CP} \right] \bmod T_{CP} - \left[\frac{\mathrm{k} + k_1}{P} T_{CP} \right] \bmod T_{CP} \right]$$

And eventually: $\delta = \varepsilon_0 - \varepsilon_1$

The error on the averaged measure is then $\varepsilon_0 - \varepsilon_1$ in $\left]-\frac{T_{CP}}{P}..\frac{T_{CP}}{P}\right[$, leading to a normalized mesure error (after multiplication by P) in $\left]-1..1\right[$, or a normalized phase error in $\left]-\frac{1}{N}..\frac{1}{N}\right[$. This error is the same than for DDMTD.

So, with some primality conditions, the ADMTD produces the same results than DDMTD when no jitter. The improvement relies on the glitch-free running through jitter zone hopping. Moreover, the

averaging operates on the jitter too, with no non-linear processing. This leads to better convergence (or at least more predictable) to « real » phase value when averaging numerous measures.

3. Special cases, limits and implementation tricks

With $P=1$, back to the DDMTD…

The maximum value of P is limited in the above description by the basic measures averaging, which needs a centering value and not too dispersed basic measures values. In a typical implementation, the beat period should be such that it always covers at last 4 successive sampling points, ie $\frac{T_{beat}}{T_{CP}}>4$, or $P<N/4$.

The minimum acquisition time T_M to insure M basic measures is :

$$T_M=(M+1)T_{beat}=(M+1)\frac{N+P}{P}T_0=(M+1)\frac{N}{P}T_{CP}$$

The ADMTD requires one multiply by P (modulus N) of the basic counter measures. The implementation is easier by choosing $P=2^p\pm1$ and $N=2^n$ (which are obviously relatively prime). The multiplier can then be reduced to one shift and one add (or sub) over n bits.

The final divide by M of the accumulated basic measures (one or less every $(M+1)\frac{N}{P}T_{CP}$ period) can be done through software or direct implementation. The particular case $M=2^mP$ with $P=2^p\pm1$ can be very efficient, by noting :

$$\frac{1}{M}=\frac{2^{-m}}{2^p-1}=2^{-m}\left(\frac{1}{2^p}+\frac{2^{-p}}{2^p-1}\right)=2^{-m-p}\left(1+2^{-p}+2^{-3p}+2^{-4p}+...\right)$$

$$\text{or } \frac{1}{M}=\frac{2^{-m}}{2^p+1}=2^{-m}\left(\frac{1}{2^p}-\frac{2^{-p}}{2^p+1}\right)=2^{-m-p}\left(1-2^{-p}+2^{-3p}-2^{-4p}+...\right)$$

With $p\geq7$, for exemple, a 32-bit precision can be achieved with 3 additions.

Summarizing, for a typical implementation of ADMTD in FPGA (no multiplier):

Let $N=2^n$.

Let $P=2^p\pm1$ with $p<n-2$, and $T_0\frac{P}{N}$ greater than the peak-to-peak jitter (f_{CP} clock + input signal).

Let $M=2^mP$.

Use $m>0$ for further jitter averaging (same result than for $m=0$ if no jitter)

4. The K-ADMTD variant

An oversampling variant can be trivially derived, by using a sampling frequency $f_{CP}=f_0\frac{KN}{N+P}$, with $N+P$ and K relatively prime. This method is called here K-ADMTD. It uses K ADMTD

subdesigns in parallel working at $f_{CP}' = f_{CP}/K$ frequency, each provided every K f_{CP} cycles with a f_{CP}-sampled bit.

In others words, the K ADMTD subdesigns operate with a helper frequency $f_{CP}' = f_{CP}/K$ and input signals shifted with a step $T_{CP} = \frac{1}{f_{CP}} = \frac{T_0}{K}\left(1+\frac{P}{N}\right)$. The final mesures of the K ADMTD subdesigns are averaged to provide the final result. Relative to one of the ADMTD subdesigns, both the resolution and the error are improved by a factor K.

The K-ADMTD is mainly useful for high f_{CP} values (more than a few 100 MHz), allowing parallelization of slower logic ADMTD blocks through a deserializer.

Summarizing, for a typical implementation of K-ADMTD in FPGA (no multiplier):

Let $N = 2^n$.

Let $P = 2^p \pm 1$ with $p < n-2$, and $T_0 \frac{P}{KN}$ greater than the peak-to-peak jitter (f_{CP} clock + input signal).

Insure $N+P$ and K relatively prime.

Let $M = 2^m P$.

Use $m > 0$ for further jitter averaging (same result than for $m = 0$ if no jitter)

In a twin-track approach, the ADMTD can also be used in subsampling mode ($f_{CP} \approx f_0/K$). We have then $T_{CP} = KT_0\frac{N+P}{N} = T_0\left(K+\frac{KP}{N}\right)$, the phase resolution is kept if K and N are relatively prime, and the jitter condition becomes : $T_0\frac{KP}{N}$ greater than the peak-to-peak jitter (f_{CP} clock + input signal).

References :

[1] P.Moreira and I.Darwazeh, "Digital Femtosecond Time Difference Circuit for CERN's Timing System", University College London

[2] D.W. Allan and H. Daams, « Picosecond time difference measurement system », 29th Annual Symposium on Frequency Control. 1975, pp 404 – 411

Legal stuff :