PLL USAGE IN THE GENERAL MACHINE TIMING SYSTEM FOR THE LHC

P. Álvarez, D. Dominguez, J. Lewis, Q. King, J. Serrano, B. Todd,
CERN, Geneva, Switzerland

Abstract

Analogue PLLs have been successfully used for decades to recover clocks and clean the jitter introduced by transmission media. Nevertheless the design parameters are hard to change once the PCB has been mounted. Digital PLLs overcome this problem. They can be either completely digital, substituting the VCO by a Numeric Oscillator, or they can keep a VCXO in case a low jitter is needed. This paper describes both configurations and gives lab results for the latter. This architecture will be used in every General Machine Timing receiver card for the LHC.

INTRODUCTION

Analogue Phase Locked Loops (PLLs) have been successfully used for decades to recover clocks and clean the jitter introduced by the transmission media.

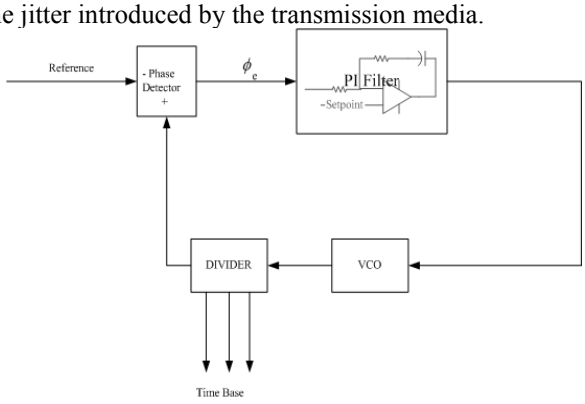

Figure 1: Generic PLL

PLLs are traditionally implemented using a digital Phase Detector whose output is low-pass filtered and fed to a Proportional Integral controller. The PI controller drives a Voltage Control Oscillator (VCO) in such a way that the estimated phase difference ϕ_e will equal the PI fixed setpoint, see Figure 1. The overall system can be seen as a jitter filter. If the PLL bandwidth is smaller than the Reference Clock jitter, the PLL will produce a Time Base with a lower jitter.

However the design parameters are hard to change once the PCB has been mounted. The extensive use of FPGAs allows us to implement fully programmable PLLs, easily changing parameters such as the lock-in time, frequency bandwidth, lock-in frequency, etc. These are additional benefits of an FPGA based PLL. Especially interesting are the possibility to adapt the phase detection algorithm to the transmission encoding scheme and the small drift after losing the reference input. In addition the PI Controller variables can be read through a bus giving the opportunity to use a software task to monitor the state of the PLL remotely.

A first approach to this structure uses a Numeric Oscillator (NO), where a down counter is programmed dynamically to compensate the variations of the Local Quartz Oscillator (LQO). We will refer to this architecture as All Digital PLL (ADPLL). As there is no control on the LQO phase, the minimum jitter will be of 1 LQO period (See Section 2). The jitter can be reduced if we use a Voltage Controlled Crystal Oscillator (VCXO). The final implementation of this configuration, which we call Hybrid PLL (HPLL), will be discussed in section 3.

ALL DIGITAL PLL

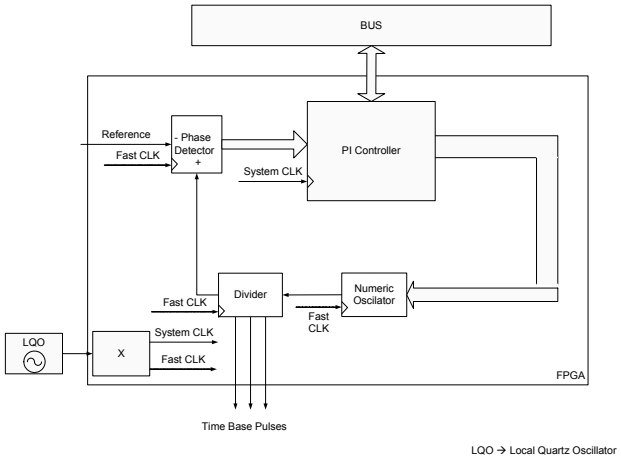

Figure 2: ADPLL architecture

The ADPLL building blocks can be seen in Figure 2: A Phase Detector (PD), a Proportional Integral (PI) Controller, a Frequency Divider and the already mentioned NO.

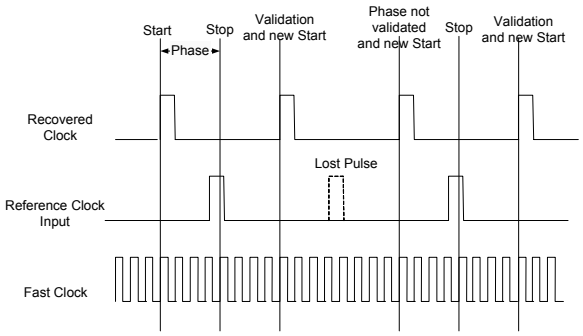

Figure 3: Phase Detector timing diagram

The Phase Detector

The PD estimates the phase between the Reference Clock input rising edge and the Recovered Clock rising edge. If there is a lost Reference Clock rising edge the PD simply doesn't do anything; this way no error is transmitted to the PI controller.

If the Reference Clock input is relatively fast (>100KHz) a decimation filter can be easily chained to reduce the high frequency noise introduced by the Fast Clock sampling. In practice this is done by accumulating in a register the direct phase estimation during a fixed number of samples.

Notice that using decimation will allow the user to fix a setpoint with granularity smaller than the Fast Clock period (T_{FC}), although the Recovered Clock jitter will obviously not be less than T_{FC}:

$$Grainularity_{PDAv} = T_{FC}/N_{Av}$$

The decimation factor (N_{Av}) can be programmed by the bus.

Proportional Integral Controller

The PI controller block is the digital translation of the classical analogue PI Controller. The calculations are performed in 32 bits fixed point adders and multipliers. These are implemented in signed logic and take into account the possibility of overflow. Several parameters can be programmed through the bus such as:

- The decimated phase setpoint
- The proportional gain, Kp
- The integral gain, Ki
- The integrator initial value

The Numeric Oscillator

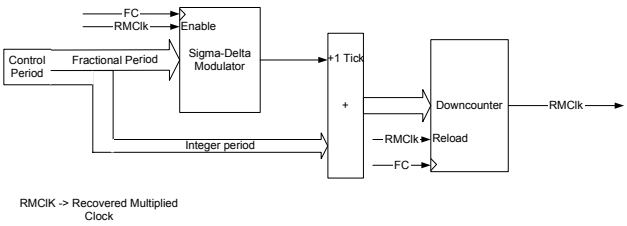

Figure 4: Numerical Oscillator Layout

The PI controller passes to the NO the period of the next Recovered Multiplied Clock (RMClk). This is divided into an integer part and a fractional part. The integer part contains the bits needed to generate a period equal to the nominal, in case the Fast Clock had no frequency error. The fractional part is fed to a Sigma-Delta modulator that provides a 1-bit signal whose average over time is equal to the fraction of Fast Clock ticks needed. For example, if we have a 40,33…MHz FC and we want to generate 1MHz, we will load 40 ticks in the Integer Period and the binary representation of 1/3 in the Fractional Period. The Sigma-Delta Modulator will add in average 1 FC tick to the downcounter every 3 RMClk pulses. The result is a clock that over the long term has a frequency of 1MHz but a short-term jitter of 25ns. The theoretical frequency control resolution is given by:

$$Grainularity_{NO} = T_{FC}/2^{NdecimalBits}$$

In case there is a large number of bits the final precision will be determined by the sigma-delta modulator noise rejection and the FC stability. In any case the RMClk peak-to-peak short-term jitter will never be less than T_{FC}.

HYBRID PLL

The HPLL structure uses a Digital to Analogue Converter (DAC) to control the frequency of a VCXO.

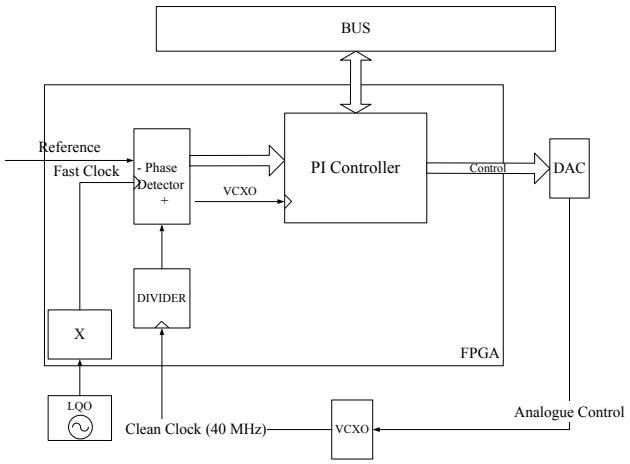

Figure 5: Hybrid PLL

Figure 5 shows the topology of an HPLL. The PI controller block is the same as the one used in the ADPLL. Its control signal is transformed into the analogue domain thanks to a DAC. The Clean Clock given by the VCXO is used to directly clock a divider to obtain an almost jitter free Recovered Clock.

The Phase Detector uses the same algorithm as in the ADPLL. In the ADPLL, FC is divided to obtain the Recovered Clock; the PD is also clocked by FC to count the number of FC ticks between the rising edges of the Recovered Clock and the Reference Clock.

Counting the number of VCXO ticks between the PD Start and Stop signals would pose a problem if we wanted to obtain a short-term jitter smaller than T_{VCXO}. Suppose we have controlled the VCXO frequency in such a way that the frequency error is null between the Recovered and Reference Clock. The PD will not be able to resolve the position of the Reference Clock between two rising edges of the VCXO clock. After this reasoning it appears clear that the phase would always drift slowly in this T_{VCXO} window after lock-in.

The problem arises because the phase measurement errors will be very correlated, whereas the effectiveness of removing errors by averaging is based on their lack of

correlation. To obtain uncorrelated errors we can act in two ways:

1. Multiply the VCXO clock until T_{VCXO} is lower than the Reference Clock jitter.
2. Use an independent stable free running oscillator to clock PD.

Option 2 has been chosen, as we cannot readily control the amount of jitter in the Reference clock.

RESULTS

The HPLL has been developed and tested in the lab. We have implemented it in a Spartan II XC2S150E FPGA. The FPGA can be accessed via VME to program the HPLL parameters. The PLL can also interrupt the VME Bus, allowing us to perform accurate tracing of the HPLL status over time.

The PLL has been configured recover a 1MHz signal by locking to a 500kBit/s Manchester-encoded data stream. The clock used to encode the data comes from a very stable rubidium oscillator. This architecture will be used in every General Machine Timing receiver card for the LHC, and the other CERN accelerators.

Figures 6 and 7 show the lock-in process for a slow HPLL and a fast HPLL configuration. The Damping Ratios have been calculated using a simple linear analogue model that does not take into account the low-pass filters introduced by the VCXO and the PD, and of course, the non-linearity related to any HPLL.

In Figures 6 and 7, the first column represents the phase error, in ns. The second column shows the output of the integrator in ppm and the third one shows the control value sent to the DAC. Each of these columns contains three rows. Row 1 shows a full-scale plot from 0 to 30s. Row 2 is a time zoom of the first 100ms, and row 3 is a vertical zoom of the last 10s.

For the Slow HPLL a jitter of less than 90ps rms has been measured for an input jitter of 180ps rms. The free running drift is less than 100ns/s (0.1 ppm) for one hour for a damping ratio of 250, a decimation factor of 2000 and a PD fast clock of 160MHz. The HPLL under test was implemented using a ±50ppm pull range VCXO.

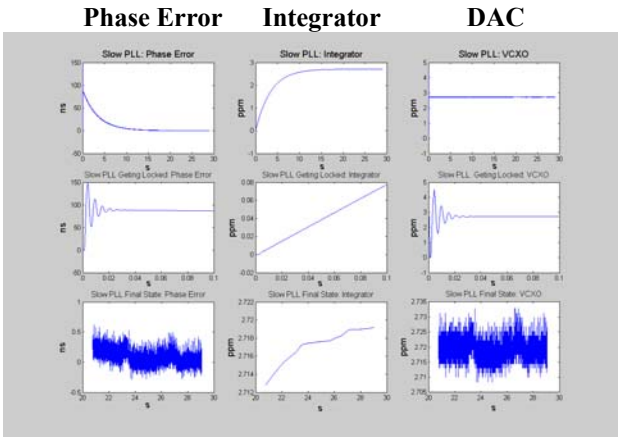

Figure 6. Slow HPLL. Damping Ratio = 250, N_{Av} = 2000, FC = 160MHz

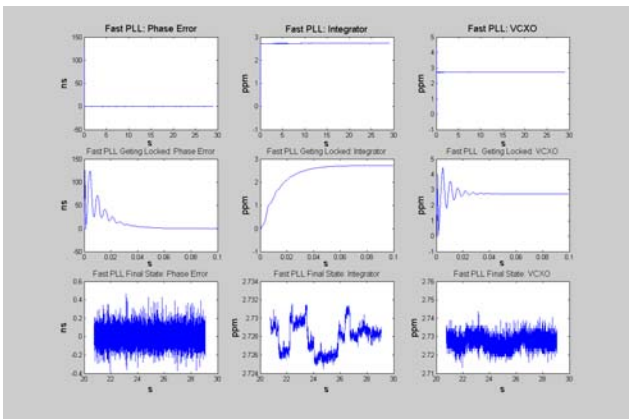

Figure 7. Fast HPLL. Damping Ratio = 16, N_{Av} = 2000 FC = 160MHz

CONCLUSIONS

We have described the architecture of the PLLs we plan to implement in the CERN General Machine Timing. The preliminary results presented here are quite promising.

REFERENCES

[1] Quentin King, Jitter measurement over World Fip, June 2000, CERN, Geneva, Switzerland. http://lhc-proj-timwg.web.cern.ch/lhc-proj-timwg/minutes/wfip_synch.PDF

[2] Ben Todd, IRIG/AFNOR Card, page 105, September 2002, CERN, Geneva, Switzerland http://btodd.home.cern.ch/btodd/xml/ab-co-ti-proj-tit.htm

[3] J. Serrano, P. Álvarez, D. Dominguez, J. Lewis, Nanosecond level UTC timing generation and stamping in CERN's LHC, ICALEPCS-03 Korea

[4] J. Lewis, Jean-Claude Bau, Javier Serrano, David Dominguez, Pablo Alvarez, The evolution of the CERN SPS timing system for the LHC era, ICALEPCS-03 Korea