COMPSYS 701 - IRP ADC-ASP

Franklin O’Sullivan

Introduction

Accurate measurement of power system frequency is a critical aspect in the monitoring and control of electrical grids. Traditional analog methods, which rely on period measurements of AC voltage, often struggle with noise and higher-order harmonics, making them less reliable. Digital methods offer improved accuracy but can be computationally intensive and slow. A new method for frequency measurement has been introduced, which is resilient to unwanted frequency components and random noise [1]. A fundamental component of this method is the Analog-to-Digital Converter (ADC), which needs to sample data accurately for subsequent digital processing.

In this project, we focus on the ADC component of this frequency measurement system. The ADC is tasked with sampling a pre-generated waveform stored in Random Access Memory (RAM) at a specific rate, allowing for precise data collection that can be easily accessed by other system components. This project aims to implement an ADC Application Specific Processor (ASP) that can interface with a Network on Chip (NoC) bus to facilitate efficient and accurate frequency measurement.

Objectives

The ADC component of this new method for frequency measurement of the power system is used to collect an ADC reading from ROM at a specific sampling rate. This ADC reading is used by other components within the frequency measurement system. For this reason, data has to be accurate and easy to access whenever it is requested. The System Clock Frequency is fixed at 100 MHz, whereas the ADC values are only available every 16 KHz. To address this issue, samples can only be read at a specific sampling rate. This rate is calculated as shown in Equation 1.

$$Sampling\ Rate\ =\ System\ Clock\ Frequency\ (100\ MHz)\ /\ (ADC\ Sampling\ Delay\ +\ 1)$$

Equation 1: ADC Sampling rate calculation

The value of the ADC Sampling Delay is user defined to ensure that ADC readings occur without error. A sufficient value for this parameter is 6249. Calculations for this value can be found in Appendix A - Sampling Rate Calculation.

In addition to periodic sampling, reconfigurability is a key objective of the ADC Application Specific Processor (ASP). It must be capable of dynamically adjusting its data width to accommodate varying precision requirements. By using the NoC configuration messages, the ADC-ASP should support data widths of 8, 10 or 12 bits, enabling seamless integration into diverse system architectures while optimising resource usage and performance.

Background Information

Designing a method in which the frequency of an AC signal can be measured both quickly and accurately has been a common problem approached by many researchers. Some of these methods have been implemented purely in embedded software through the use of microcontrollers; however, they suffer in terms of accuracy or speed. Instantaneous frequency measurement brings disadvantages. Due to the short observation period, impulsive changes in the input magnitude or phase can appear as large excursions in frequency. They are further susceptible to noise, requiring careful consideration of filtering requirements [2]. Other methods, such as Fast Fourier Transform (FFT) and Taylor series approximation, have also been used, though these suffer in terms of noise or speed.

A new method for frequency measurement of the power system was proposed, having been derived from an algorithm developed for symmetry detection. This method of symmetry detection has been used for cycle detection in sinusoidal signal corrupted by unwanted noise, including DC component, higher-order harmonics and random noise. This algorithm detects the location where only the positive part of a power system waveform is maximally symmetric, substituting this for a zero-crossing. Using cross-correlation to help remove random noise in the input signal, the frequency and the rate of change of frequency can be measured accurately. This method is proposed to be not only quick, occurring in a single clock cycle, but also accurate due to the relatively complex computation.

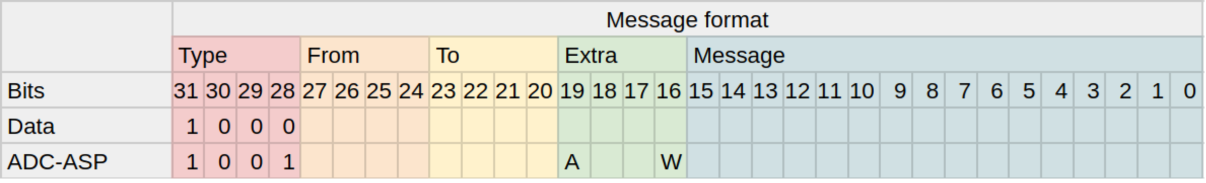
A component of this new method of frequency measurement is ADC. Data needs to be sampled from the environment at a reasonable rate to ensure that subsequent calculations will be relevant to the real world. The data needs to be transferred to other components in a time-predictable manner. The interconnection network, Network on Chip (NoC), allows for inter-core communication in systems with many cores. A time-predictable NoC interconnect called TDMA-MIN combines the Time Division Multiple Access (TDMA) scheme with the Multistage Interconnect Network (MIN) in a novel way with fixed bounds on communication cost for any number of cores or nodes [3]. This method of communication allows the ADC component to easily share data with other components at it’s designated sampling rate.

Achievements

The ADC-ASP was successfully completed as described in the requirements of the project. The ADC-ASP is able to read data from a RAM memory block at the set sampling rate. This data can then be put on the NoC intercommunication bus so the data can be used in subsequent components.

The ADC-ASP successfully reads data from RAM memory at the specified sampling rate of 16 MHz, ensuring precise capture of the analog signal for subsequent processing. This consistent sampling is vital for maintaining data integrity and system reliability.

Configurability was a further requirement of the AD-ASP. This was achieved through the use of NoC configuration messages. These messages could be sent to a specific component at any time to change or modify how data is handled. A simplified NoC message format can be seen in Figure 1.



	Message format																															
	Type				From				To				Extra				Message															
Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Data	1	0	0	0																												
ADC-ASP	1	0	0	1									A			W																

Figure 1 Simplified NoC message format

The message can be split into five sections: message type, sender, receiver, extra and message. The first four sections are all 4 bits each. The final section is 16 bits, leading to a message length of 32 bits.

The first section describes the type of message. Messages received by the ADC-ASP can be either data messages with the code “1000” or configuration messages with the code “1001”. This section of the message dictates how the NoC message will be handled by the ASP. Generally, the ASP will be responsible for sending data messages, so these do not need to be received. A configuration message, on the other, is used to change the send address (A) or the data width (W). By setting one of these flags high at a time, the ASP can read the message and use the value found as its new send address or data width.

Methodology

Timing

A key component of designing the ADC-ASP was ensuring that the datapath was organised correctly to read data from the memory and share it on the NoC bus. Data is read from (RAM), which has been preconfigured with data from a .mif file. This data is read at a sampling rate of 16 MHz to get rid of any aliasing effect while retaining an effective measurement.

To achieve the correct sampling rate, a counter is used. This counter ensures that the data is read from the RAM at the correct sampling rate. Because the system clock is 100 MHz, a counter of 6249 is required. When this counter reaches 6249, it will make a successful comparison within a comparison block. This will reset the value as well as fetch new data from the memory and send it on the NoC bus.

Configurability

To accommodate different applications and improve the versatility of the ADC-ASP, the output data width can be configured. This configurability allows the ADC-ASP to support multiple memory widths, enhancing its utility. The default width is 12 bits wide. This is achieved by always using a 12-bit wide RAM memory block and filling the unused bits at the end of the number with zeros when 10-bit or 8-bit values are being read from the memory. This flexibility allows for lower precision and, therefore, faster calculations. The data width configuration is controlled via NoC messages, with bit 16 being set high to indicate that a new data width is requested.

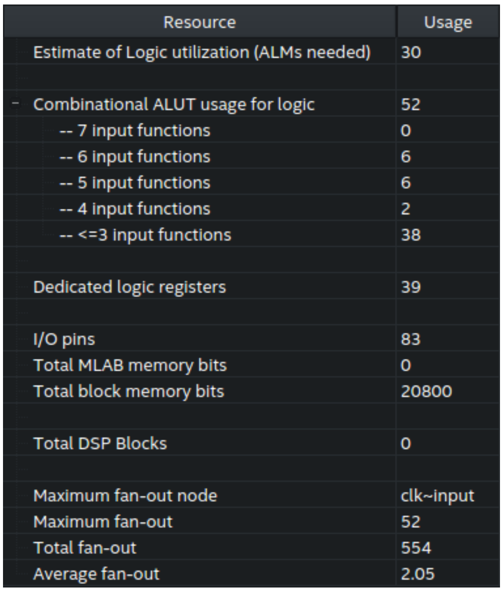
Another aspect of the ADC-ASP that can be configured is the address of the next component in NoC messages. This can be configured in the case that a component in the method is not being used or if addresses change for any reason. To change the address of the next component, the 19th bit of the NoC message must be set high, with the new address passed in the bottom 16 bits as data.

Data Flow and Processing

When data is read from the RAM memory block at the set sampling frequency, this data needs to be shared with other components on the NoC. This is done through the use of a NoC message. This message is sent in the format of a data message, as shown in Figure 1. When the data is requested with 12-bit accuracy, this value can simply be loaded into the message section of the NoC message, with the three highest bits being set to zero. If a message of lower accuracy is sent, the top three bits will be set to zero, and the lowest two or four bits will also be set to zero.

Resource usage

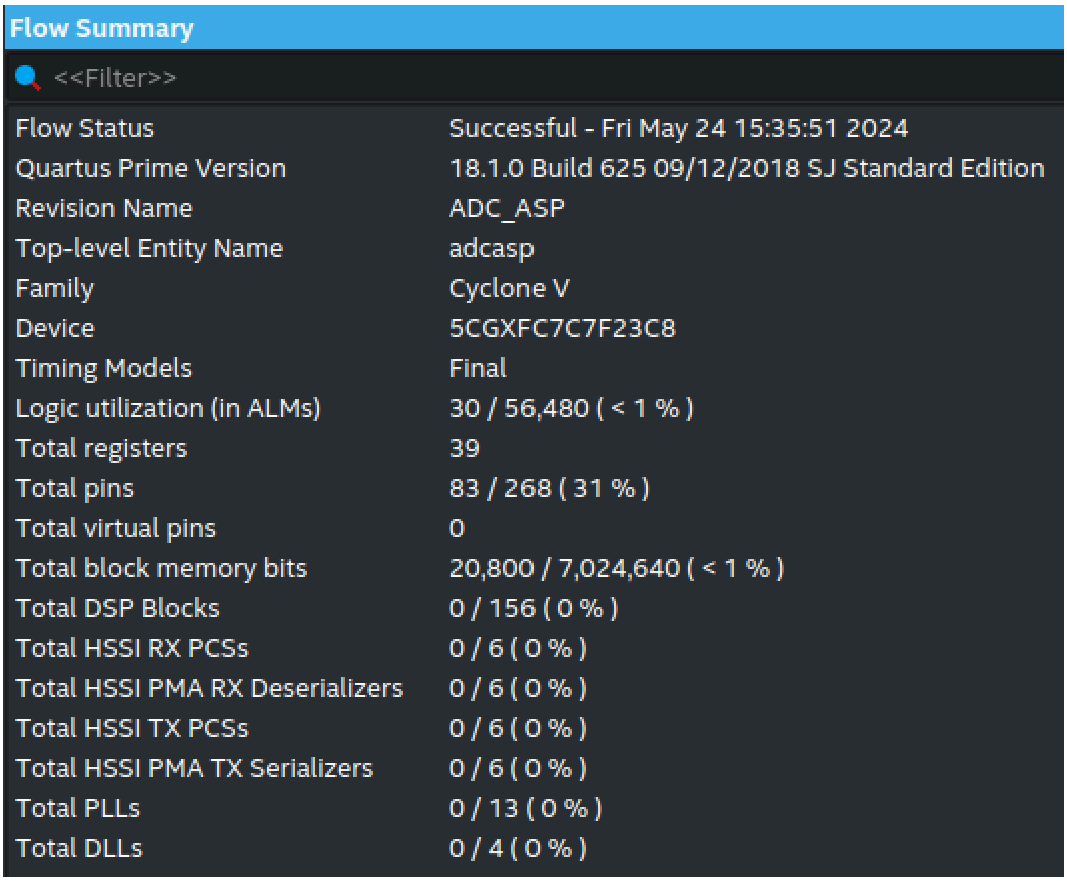
When the ADC-ASP is implemented in this manner it uses approximately 30 Adaptive Logic modules (ALM). This can be seen in the resource usage summary shown in Figure 2.



Resource	Usage
Estimate of Logic utilization (ALMs needed)	30
Combinational ALUT usage for logic	52
-- 7 input functions	0
-- 6 input functions	6
-- 5 input functions	6
-- 4 input functions	2
-- <=3 input functions	38
Dedicated logic registers	39
I/O pins	83
Total MLAB memory bits	0
Total block memory bits	20800
Total DSP Blocks	0
Maximum fan-out node	clk~input
Maximum fan-out	52
Total fan-out	554
Average fan-out	2.05

Figure 2: Resource usage summary.

The flow summary in Figure 3 further displays the logic utilisation of the ASP. This utilisation is 30 / 56,480, or less than 1%. This leaves enough space for other components to coexist on the FPGA board.

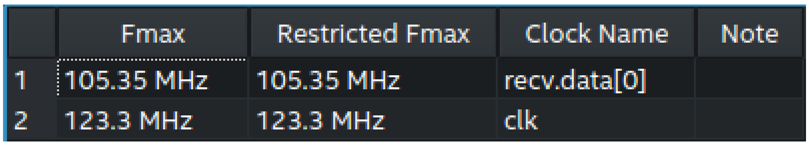


Flow Summary

Flow Status	Successful - Fri May 24 15:35:51 2024
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Standard Edition
Revision Name	ADC_ASP
Top-level Entity Name	adcasp
Family	Cyclone V
Device	5CGXFC7C7F23C8
Timing Models	Final
Logic utilization (in ALMs)	30 / 56,480 (< 1 %)
Total registers	39
Total pins	83 / 268 (31 %)
Total virtual pins	0
Total block memory bits	20,800 / 7,024,640 (< 1 %)
Total DSP Blocks	0 / 156 (0 %)
Total HSSI RX PCSs	0 / 6 (0 %)
Total HSSI PMA RX Deserializers	0 / 6 (0 %)
Total HSSI TX PCSs	0 / 6 (0 %)
Total HSSI PMA TX Serializers	0 / 6 (0 %)
Total PLLs	0 / 13 (0 %)
Total DLLs	0 / 4 (0 %)

Figure 3: Flow summary of ADC-ASP

Another important point in the design of the ADC-ASP is that the processor is able to work with the allocated clock frequency of 100 MHz. In Figure 4, it is shown that the maximum possible clock frequency for the ASP is 105.35 MHz. This fits with the requirements and shows that the ADC-ASP will be able to effectively work with other components in the same system.

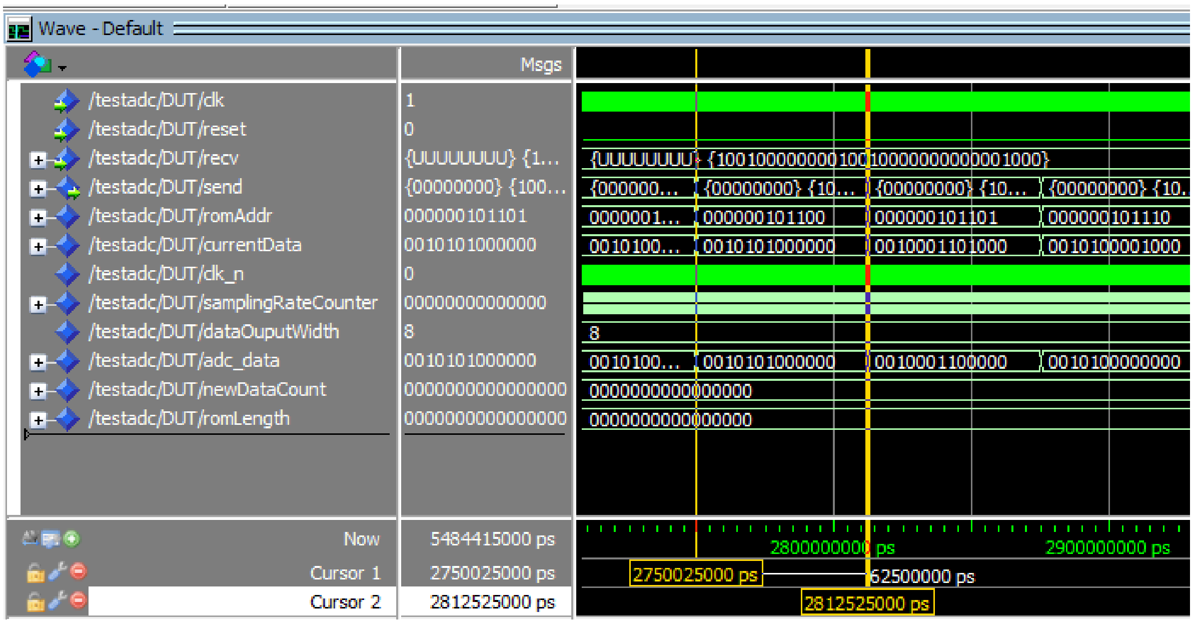


	Fmax	Restricted Fmax	Clock Name	Note
1	105.35 MHz	105.35 MHz	recv.data[0]	
2	123.3 MHz	123.3 MHz	clk	

Figure 4: Maximum clock frequency of the ADC-ASP

Results

The sampling rate of the ADC is requested to be 16 KHz, corresponding to an expected interval of 62.5 us between samples. This was verified using Model Sim simulations, as shown in Figure 5. The timing analysis confirms that the interval between ADC samples is consistently 62.5 us, or 62500000 ps, demonstrating that it accurately maintains the desired sampling rate.

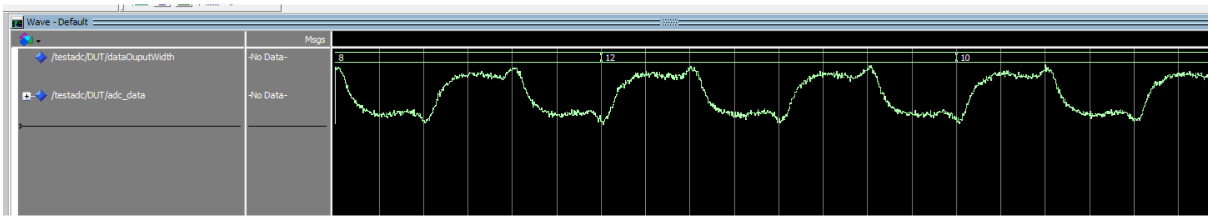


Wave - Default

	Msgs
/testadc/DUT/clk	1
/testadc/DUT/reset	0
/testadc/DUT/recv	{UUUUUUUU} {1...
/testadc/DUT/send	{00000000} {100...
/testadc/DUT/romAddr	000000101101
/testadc/DUT/currentData	0010101000000
/testadc/DUT/clk_n	0
/testadc/DUT/samplingRateCounter	00000000000000
/testadc/DUT/dataOuputWidth	8
/testadc/DUT/adc_data	0010101000000
/testadc/DUT/newDataCount	0000000000000000
/testadc/DUT/romLength	0000000000000000
Now	5484415000 ps
Cursor 1	2750025000 ps
Cursor 2	2812525000 ps

Figure 5: Sampling period verification

The ADC-ASP supports configurable data widths of 8, 10 or 12 bits, allowing it to adapt to different precision requirements. Based on input messages from a test bench, the ADC-ASP successfully reconfigures the data width. As illustrated in Figure 6, the output data on the NoC bus accurately represents the input waveform for all three configurations. This ensures that the system can flexibly trade-off between data precision and resource usage without compromising the integrity of the sampled waveform.



Wave - Default

	Msgs
/testadc/DUT/dataOuputWidth	-No Data-
/testadc/DUT/adc_data	-No Data-

Figure 6: Input waveform for different bit widths

The ADC-ASP implementation utilises less than 1% of the available ALMs on the FPGA board, as depicted in Figure 4. The design utilises approximately 30 ALMs out of the 56,480 available. This low resource usage indicates that the ADC-ASP is highly efficient and leaves ample space for additional components and processes.

Discussion

The results obtained from the ADC-ASP implementation and simulation provide several insights into its performance and efficacy in power system frequency measurement applications.

Modelsim simulations have verified that the ADC-ASP accurately maintains the desired sampling rate of 16 KHz for all samples. This precise timing is crucial for the reliable measurement of power system frequencies, ensuring that the sampled data represents the actual signal without introducing aliasing or significant timing errors.

The ADC-ASP's capability to support multiple data widths (8, 10 and 12 bits) is a significant advantage. This flexibility allows the system to adapt to different precision requirements and resource constraints. As shown in Figure 6, the output data for the NoC bus correctly represents the input waveform across all configurations. This adaptability can help optimise the balance between computational load and accuracy.

The implementation of the ADC-ASP utilises less than 1% of the available ALMs on the FPGA board. With approximately 30 out of 56,480 being used. This leaves ample room for additional components and processors to be integrated into a larger system. This makes it suitable for complex power measurement systems requiring multiple processing units.

Conclusion

The development and implementation of the ADC-ASP has demonstrated successful maintenance of a precise sampling rate, as well as flexibility when working with configurable data widths. The system is able to reconfigure data widths without compromising the integrity of the sampled waveform, as verified through extensive simulations. The adaptability is beneficial for optimising performance based on specific application needs.

Overall, the ADC-ASP provides a robust and efficient solution for power system frequency measurement, using less than 1% of the available resources on the designated FPGA board. The integration of the ADC-ASP onto the NoC bus further enhances its applicability in real-time systems, specifically for accurately measuring the frequency of an AC power signal.

References

[1] Z Salcic, R Mikhael, 2000. A new method for instantaneous power system frequency measurement using reference points detection, Electric Power Systems Research, Volume 55, Issue 2, 1 August 2000, Pages 97-102, https://doi.org/10.1016/S0378-7796(99)00102-9,

[2] Moore, P.J., J.H. Allmeling, and A.T. Johns. "Frequency Relaying Based on Instantaneous Frequency Measurement [Power Systems]." *IEEE Transactions on Power Delivery* 11, no. 4 (October 1996): 1737–42. https://doi.org/10.1109/61.544251.

[3] Salcic, Zoran, Muhammad Nadeem, and Bjoern Striebing. "A Time Predictable Heterogeneous Multicore Processor for Hard Real-Time GALS Programs." In *ARCS 2016; 29th International Conference on Architecture of Computing Systems*, 1–8, 2016. https://ieeexplore.ieee.org/document/7499233.

Appendix A - Sampling Rate Calculation

$16KHz \ = \ 100MHz \ / \ (ADC\ Sampling\ Delay \ + \ 1)$
$100MHz/16KHz \ = \ ADC\ Sampling\ Delay \ + \ 1$
$6250 \ = \ ADC\ Sampling\ Delay \ + \ 1$
$ADC\ Sampling\ Delay \ = \ 6250 \ - \ 1$

$ADC\ Sampling\ Delay \ = \ 6249$