HDL-Based Implementation of a Novel Dynamic Reconfigurable Data Acquisition Scheduler-on-Chip (SchoC)

M. Abdallah
State University of New York IT
abdallm@sunyit.edu

Abstract

Data acquisition (DAQ) is a crucial component in instrumentation and control. It typically involves the sampling of multiple analog signals and converting them into digital formats so that they can be processed. DAQ systems also involve microprocessors, microcontrollers, digital signal processing, and/or storage devices. Many multi-channel DAQs, which utilize some sort of processing for simultaneous input channels, are found in various applications. In this research for heterogeneous multi-channel signals, different sampling rates are identified for each channel and optimized for best data quality with minimal storage requirement. Accordingly, power consumption and transmission times can be reduced. Moreover, the proposed system is dynamic, which means it detects any change in any of the input channels/signals. It also adapts to the change and tries to find the optimal sampling rate for each signal on the fly. The fidelity of the proposed scheduler-on-chip (SchoC) is increased by using reconfigurable chip technology, where flexibility, concurrency, speed, and reconfiguration can be achieved in hardware. Therefore, SchoC can be utilized in various real world applications, especially hazardous environments, or for remote architecture reconfiguration, while keeping the cost of the device low. Performance evaluations show that the speed of the SchoC is 24% faster than a comparable software-based scheduler. The SchoC reduces the amount of data acquired by up to 59%, which in turn decreases memory requirements. It achieves a high signal reconstruction quality. It reaches up to 15% better root mean square of errors than a comparable software-based DAQ.

Introduction

In most real-time systems, task scheduling is the most important problem, because it is the scheduling plan that ensures that tasks meet their deadlines. Each task has its own timing restrictions or deadlines. Existing scheduling algorithms can be classified as shown in Figure 1. Dynamic scheduling is more flexible, allowing characteristic-varying tasks to be scheduled. In Round-Robin, processor time is equitably divided among all processes. Each process gets an equal time slot of a shared processor. If heterogeneous multi-rate processes are scheduled by a single core processor, Round-robin scheduling technique assigns the shared processor to all processes with a fixed processing rate [1]. In rate monotonic (RM), tasks are assigned different priorities. Tasks with higher priority will interrupt the current task and replace it. This also means that the system is preemptive. The priorities are assigned to tasks based on their frequency. Priorities are assumed to be static, so the task periods also need to be static. Hence, RM cannot be used if tasks have varying frequency [2]. Earliest deadline first (EDF) places processes in a priority queue. The process that is closest to its deadline will be scheduled for execution. It has some drawbacks; for example, in situations where deadlines are not known in advance, they are provided but subject to change and/or situations that require uniform process spacing [2]. Scheduler-on-chip (SchoC) is a new technique to dynamically schedule a large number of heterogeneous multi-rate tasks on a single core processor without increasing its requirements. It can fit under the quality-driven item in Figure 1.

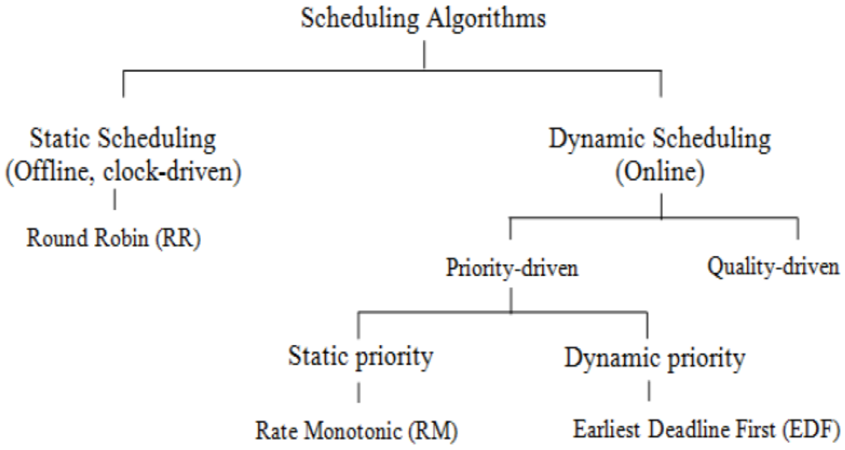


Figure 1. Classification of existing scheduling

The contribution of this research is to propose a low cost, small circuit size data acquisition system for a multiple channels system where several applications must schedule their tasks on a single processor. Each of the several channels requires its own optimal sampling rate. Therefore, the problem of scheduling the tasks can be formulated as an optimization problem.

Given a maximal sampling rate F_s of an analog-digital converter (ADC) for a total number of N channels, the problem will be how to assign an optimal sampling rate F_{si} for each channel. A mathematical formulation of the problem is proposed. This algorithm allows maximizing the total number of channels but also minimizes the amount of data that is acquired, stored, and eventually transmitted. The system is prototyped on an Altera Cyclone II field programmable gate array (FPGA). The scheduling algorithm that finds the optimal sampling rate is developed using only hardware description language (HDL) modules (i.e., fully hardware) and compared against the known scheduling techniques (Round-Robin, RM, and EDF). The speed of the proposed scheduler is compared against software-based scheduling implementation on the NIOS II processor using C++. The power consumption calculations and comparison is currently under investigation and will be reported in future steps.

Applications

Using a single ADC is the best solution in terms of circuit size and power consumption compared to a dedicated ADC per channel. Using a single ADC to acquire multichannel is the optimal solution in nanotechnology applications since the circuit size is a critical issue. The proposed SchoC can be utilized in many applications since different applications are needed to schedule multi-rate tasks on a single core processor. The SchoC can be used in a low cost, high performance multi-channel DAQ system for custom sensing applications that have a matrix of different analog sensors. Alongside with analog sensors (voltage, current, or charge output), quasi-digital sensors and transducers (frequency, period, PWM output) can use the SchoC [3]. Moreover, in environmental measurements, the operational satellite synthetic radar was used to acquire different data at different frequencies in order to estimate the soil moisture in different sites on the earth [4]. The system needs to acquire many signals within a short period of time due to uncontrollable changing site conditions. In addition, it can be used in a versatile instrument that can be the basis of developing a spectroscopic imaging [5], [6]. The SchoC can be used in order to acquire human body signals such as heartbeat, pressure, and lung sound. Using a varying sampling rate per channel is the optimal solution in terms of scalability, power consumption, and memory requirements [7], in addition to applications with only one channel of varying data sources. During each time period, sampling rate should be configured according to the input channel characteristics. In the situation of a single sensor of varying data sources, such as irradiation or vibration, the dynamic SchoC can be used as well.

Design Methodology

For dynamic heterogeneous multi-channel signals, it is proposed that each channel be sampled at the appropriate sampling frequency to maximize the total number of channels and minimize the amount of data being acquired that eventually will be stored or transmitted. The SchoC is designed and implemented in FPGA to manage the variable switching time of the ADC and the multiplexer such that an arbitrary large number of channels can be acquired with a good reconstruction quality.

One of the advantages of the proposed system that is not considered here is power consumption. According to Equation 1, the dynamic power increases as the activity factor (α) increases, which in turn will increase the total power consumption. If the sampling rate increases, the activity factor will be increased.

$$P_{total} = \alpha \, C \, V^2 \, F + I \, V + P_{short} \qquad (1)$$

where *C* is capacitance, *V* is voltage, *F* is processor clock frequency. Generally speaking, dynamic power can be reduced by using different smaller rates of activity factor for less frequent sampling for channels that require a much smaller minimum service rate as compared to channels with higher service rates.

Optimization Problem

The problem can be formulated as an optimization problem. Given a maximum sampling rate (F_s) of ADC and a total number (N) of channels, an optimized sampling rate (F_{si}) for each channel needs to be assigned.

$$\begin{aligned} & Max \sum_{i=1}^{N} \log_2 (OSR_i) \\ subject\, to: \; & \sum_{i=1}^{N} F_{si} \le F_s \\ & T_{si} \% \, T_s = 0 \\ & T_{si} \times M_1 \ne T_{sj} \times M_2 + T_s \times M_3 \end{aligned} \tag{2}$$

where oversampling ratio for channel (i) = $OSR_i = F_{si} / F_i$; i, j =1,...,N ; $i \neq j$; $T_{si} = 1/ F_{si}$; $T_s = 1/ F_s$; M1, M2, and M3 are integers (≤ 2 Max(T_{si})) ; M3 is the number of time periods (T_s) between channel (j) and channel (i).

The objective of this optimization problem is to maximize the assigned sampling rate (or minimize sampling period) for each channel. However, various restrictions limit the OSR_i. First, the summation of assigned sampling rates for all channels must be less than or equal the total sampling rate of the available ADC. Second, the assigned sampling rate for any channel cannot be an arbitrary number. Its inverse (i.e., the time period) must be a multiple of the inverse of the total sampling rate of the available ADC (T_s). Third, at any given time, no more than one channel can be sampled. For example, assume T_s is 0.625 nsec, Ch_1 has T_{s1} equal to 2.5 nsec, and Ch_2 has T_{s2} equal to 1.875 nsec, so there is a problem at time T=2.5 nsec. Both Ch_1 and Ch_2 need to be sampled at the same time. This problem is called a same time sampling (STS) problem. This situation has to be avoided in order to maintain a high signal reconstruction quality.

SchoC Implementation

This section presents the implementation of the proposed SchoC using Altera FPGA Cyclone II and considers four design issues. First, the scheduler logic is presented. Second, the software characteristic of the software implementation of the proposed system is introduced. The software implementation is only introduced for comparison purposes (i.e., to be compared with the proposed fully hardware implementation of the proposed scheduler). Third, the digital signal pre-processing unit (DSPU) is presented. The DSPU detects each channel's frequency. Finally, the frequency detection and monitoring (FREDM) module is presented. It detects any change in the frequency and reports it.

Figure 2 presents a SchoC flow chart. The inputs to the SchoC are frequency bandwidths of input channels, calculated on-the-fly by a special DSP module implemented in the FPGA using only HDL modules. Therefore, it is different than the EDF scheduler where the user/operator must input these frequency bandwidths in advance.

The SchoC technique starts with state 0 (S0) where each channel has its own sampling rate F_{si}, which equals the Nyquist sampling rate. Therefore, each channel sampling period T_{si} is calculated by taking $1/F_{si}$ in state 1 (S1). T_{si} should be multiples of T_s. If it is, the flow goes to state 2 (S2). Otherwise, T_{si} should be updated to be multiplies of T_s. In state S2, brute force search method is used. It is necessary to check the total summation of sampling frequencies of channels. If it exceeds the maximum sampling rate of the available ADC, the task will not be schedulable. Otherwise, the control goes to state 3 (S3). Current sampling frequencies could be a solution. However, the STS problem should be checked before judging the solution in hand. It comes to state 4 (S4) where channels order iterations will take place. In each order, STS problem will be checked. If there is an STS problem, a success rate will be calculated and then the control goes to S4 again for the next iteration. If there is no STS problem, we have a valid solution and it should be stored as a possible optimal solution. After storing the solution or after the whole iterations are finished without finding a solution, state 5 (S5) should be reached. In S5 for each channel, T_{si} will be decremented once by the value of Ts, taking into account that T_{si} cannot be smaller than T_s. Then the flow goes again to S2.

Software vs. Hardware Implementation

The proposed system is implemented twice. It is implemented in fully hardware and in software to compare both results. A NIOS II processor is used to implement the software solution. The processor consists of three types of cores: NIOS-II/e (economy), NIOS-II/f (fast), and NIOS-II/s (standard). The NIOS-II/f is the core used in this research. It is a 32-bit RISC processor. The processor is pipelined and has an instruction cache, data cache, and 51 Dhrystone million instructions per second. This processor is provided with a JTAG debug module that has software debug capabilities that enable a designer to debug the application when launched in hardware. NIOS II processor is based on Harvard architecture, where the program and data are stored separately. It has a 4-kB instruction cache and a 2-kB data cache. The processor runs on a 50 MHz.clock. The NIOS-II processor core is connected to its components through the Avalon switch interface.

Digital Signal Pre-Processing Unit (DSPU)

DSPU performs automatic signal conditioning such as bias addition and removal, adaptive signal scaling, and filtering proposed and developed by the author [8]. An extension of the DSPU is presented here to have the built-in capability to perform digital signal processing such as FFT and DCT for one dimensional digital signal. Its added function in this paper is to perform real-time fast Fourier transformation (FFT) for each channel to monitor each varying channel frequency. As shown in Figure 3, DSPU has two main tasks. It performs FFT for each channel. In addition, a power density calculation is performed to detect each channel frequency bandwidth (F_1, F_2 … F_N). Moreover, it passes these values to the frequency detection and monitoring (FREDM) module, which in turn automatically decides if frequency change occurs.

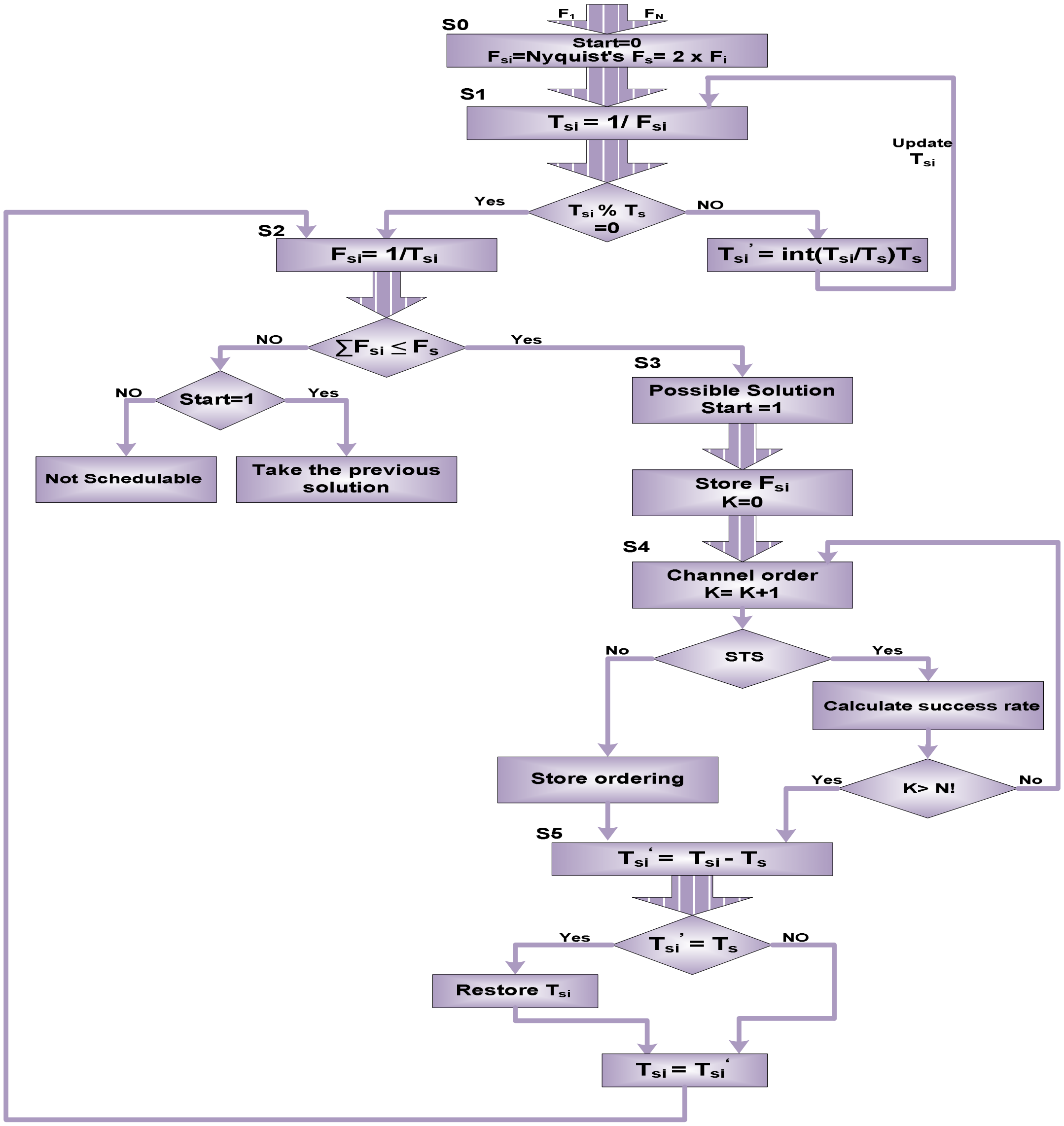


Figure 2. The proposed SchoC flowchart

In the beginning, a number of samples (M) is acquired from each channel. DSPU performs the FFT on these samples. using the Altera FFT megacore function. The FFT megacore function's outputs are M points, where M is the number of points used in FFT processing. Each point has real and imaginary parts. For each channel, M output points are applied to a power density calculation module. The power density is calculated as follows in equation 3.

$$P_i = \frac{(realpart_i)^2 + (imagpart_i)^2}{M}, i = 1, ..., M \qquad (3)$$

It is well known that only the first half of M points is used to determine the frequency bandwidth. The second half is an identical mirror of the first half. An input signal is not necessary a single frequency signal. It may have multiple frequency components. In order to detect the frequency bandwidth in this case, the greatest frequency component should be detected. The greatest frequency component will be used to decide the optimal sampling rate for a channel.

Frequency Detection and Monitoring (FREDM) Module

After detecting frequency bandwidths (F_1, F_2 … F_N) via DSPU, frequency change detector in FREDM compares each channel frequency with their respective previous values. If both values are within a preset small range, no action is taken. Otherwise, new frequency values are stored in a memory, and an enable signal is activated. The enable output signal is the input to the SchoC. Moreover, the action signal can simply report the change in frequency to a user. It indicates which channel has the frequency change. The action signal can drive a certain device to perform a predefined action such as alarm or frequency restoration (in power systems). Figure 3 illustrates the intercommunication between DSPU, FREDM, and SchoC.

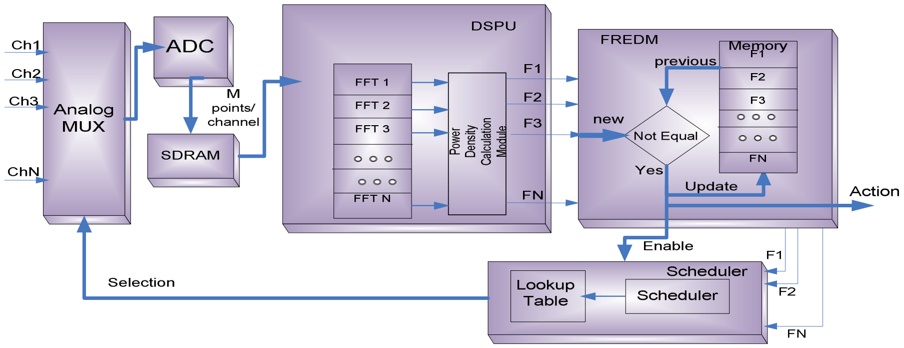


Figure 3. DSPU module, FREDM and SchoC

SchoC Evaluation

This section discusses different performance evaluation aspects. Six parameters are presented to verify the performance of the proposed SchoC. Signal reconstruction and speed are used to compare between the software implementation and fully hardware implementation of the proposed scheduler. Accuracy is used to compare between the fully hardware implementation of the scheduler and a well-known existing technology (NI). Memory requirements and schedulability analysis are compared between the proposed SchoC and the well-known Round-Robin scheduling technique. Finally, synthesis analysis of the proposed SchoC is presented.

Signal Reconstruction

Both EDF, RM scheduling, and the proposed SchoC, techniques are implemented in software using the NIOS II processor. Moreover, the SchoC is implemented again in only hardware. When the scheduler implementation is done via software processor, low quality reconstruction signals are generated, especially with high frequency signals. NIOS II processor is used to implement the proposed scheduler, EDF, and the RM scheduling techniques. Its instructions have a non-uniform execution time. In other words, the time between each acquired sample is not equal. A Tektronix logic analyzer is used to prove this observation. The logic analyzer is connected to the acquisition clock of the NIOS II processor via one pin of I/O pins of the used FPGA. Figure 4 shows the non-uniform behavior of the generated acquisition clock of the NIOS II processor. This affects the frequency of the stored signal. So, one can find that the stored reconstructed signal starts slow down and deviates from the original signal. This comes from the fact that NIOS II processor is instruction-based, which makes the data acquisition non-uniform. In other words, the time between each acquired sample is not equal. This affects the frequency of the stored signal. Therefore, the proposed hardware SchoC is better than the software-based implementation in terms of the signal reconstruction's quality.

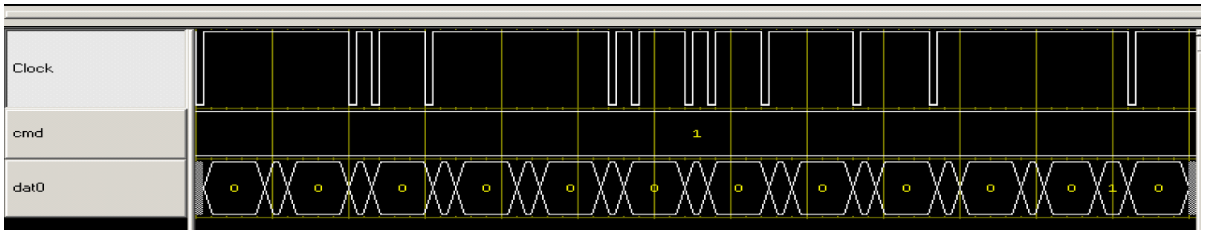


Figure 4. Logic analyzer shows the non-uniform behavior of the NIOS II processor acquisition rate

Speed

In software-based DAQ systems, the speed of data acquisition, processing, and storing is slow. It takes 46 seconds to process and store only 8000 blocks of data where each block is 512 samples. Hence, another design approach is proposed and implemented. The SchoC hardware acquisition is implemented to get faster data acquisition, processing, and storing system as well as maintain an accurate signal reconstruction in terms of the frequency.

A comparison is done between the proposed SchoC and the software-based NIOS II acquisition and storing systems. Different data sizes are applied to both systems. The performance evaluation shows that the speed of the proposed fully hardware SchoC is 24% faster than a comparable software-based scheduler. Figure 5 shows the saved time using the SchoC rather than the software-based schedulers. The horizontal axis represents the data size (number of blocks), and the vertical axis represents the saved time in seconds. As shown, as the data size increases, the saved time also increases.

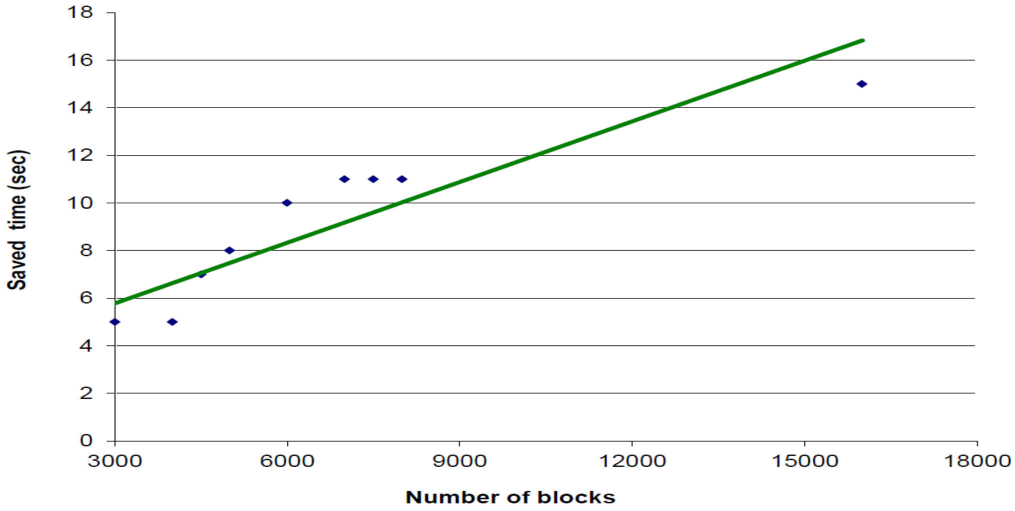


Figure 5. Saved time using the proposed SchoC

Accuracy

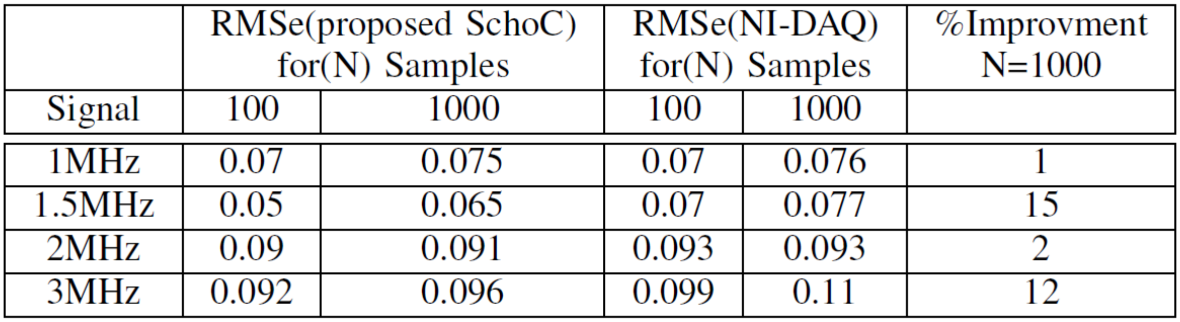
In this subsection, a comparison study between the proposed SchoC and an existing technology DAQ system is presented. The National Instrument (NI) data acquisition card is chosen because it has the closest similarity to the proposed DAQ structure (although it is a computer-based). The NI test-bench is a PCI-6115 4 channel-12 bit DAQ card [9]. National Instruments DAQ has been used as a comparison reference. It is a computer-based DAQ. For the sake of fairness, the sampling rate is fixed for both systems to be 40 MSPS. The SchoC utilizes the Texas Instruments ADC12D040 12 bit ADC [10]. Different signals have been applied to both systems. Acquired signals from both systems have been tested in terms of root mean square of errors. A signal generator has been used to generate the sine waves under investigation.

The proposed SchoC works as stand-alone, without any interference from the computer. All the processing and control are done by the FPGA. On the other hand, the NI-based DAQ needs a LabView program that runs on a computer to store the input signal into a file in the attached computer. Both acquired/stored signals by both systems have been tested by MATLAB. The root mean square of errors has been used as an evaluation parameter.

Four analog sine waves (1, 1.5, 2, and 3 MHz) signals are applied into the input of both DAQ systems. Different numbers of samples (N = 100 and 1,000) are considered. A comparison between the acquired/stored signals via both DAQ systems with respect to the source signal is done. The root mean square of errors RMS (e) is calculated for both systems.

Table 1 shows a comparison between the proposed SchoC and a comparable NI-based system. From Table 1, although the NI card has two signal conditioning stages, the performance of the proposed SchoC is better than the NI-DAQ. The best performance for the proposed SchoC has 15% better (smaller) RMS (e) than NI-based DAQ in the case of 1000 sample of 2 MHz input signal.

Table 1. Root mean square of the error for both proposed hardware FPGA-based acquisition SchoC And Ni-based DAQs



	RMSe(proposed SchoC) for(N) Samples		RMSe(NI-DAQ) for(N) Samples		%Improvment N=1000
Signal	100	1000	100	1000	
1MHz	0.07	0.075	0.07	0.076	1
1.5MHz	0.05	0.065	0.07	0.077	15
2MHz	0.09	0.091	0.093	0.093	2
3MHz	0.092	0.096	0.099	0.11	12

Memory Requirements

In this section, both Round-Robin and the proposed SchoC schedulers are only implemented in hardware. The goal is to test the ability of the proposed SchoC to minimize the memory requirements. Various case studies are considered for verification. One is presented in this section. Eight sinusoidal analog signals are considered. The frequencies of the eight input channels are 500kHz, 200 kHz, 190 kHz, 185 kHz, 160 kHz, 100 kHz, 100 kHz, and 100 kHz. Let the maximum sampling rate F_s of the available ADC be 10 MSPS (T_s = 100 nsec). The SchoC is shown in Figure 6. The x-axis represents time slots separated by T_s. The y-axis represents the acquired channels (1 to 8). On the other hand, Round-robin scheduler can schedule the given signals with a fixed sampling rate for all channels (1/800nsec = 1.25MSPS). This leads to oversampling low frequency signals (such as: the 100 kHz), which in turn causes extra memory storage requirements and power consumption as well. The SchoC schedules the eight input channels to be sampled at a varying sampling rate. This reduces the amount of data being acquired, which in turn decreases the required memory. The total amount of data being acquired using the proposed SchoC is less than that acquired by the Round-Robin scheduler by 59%.

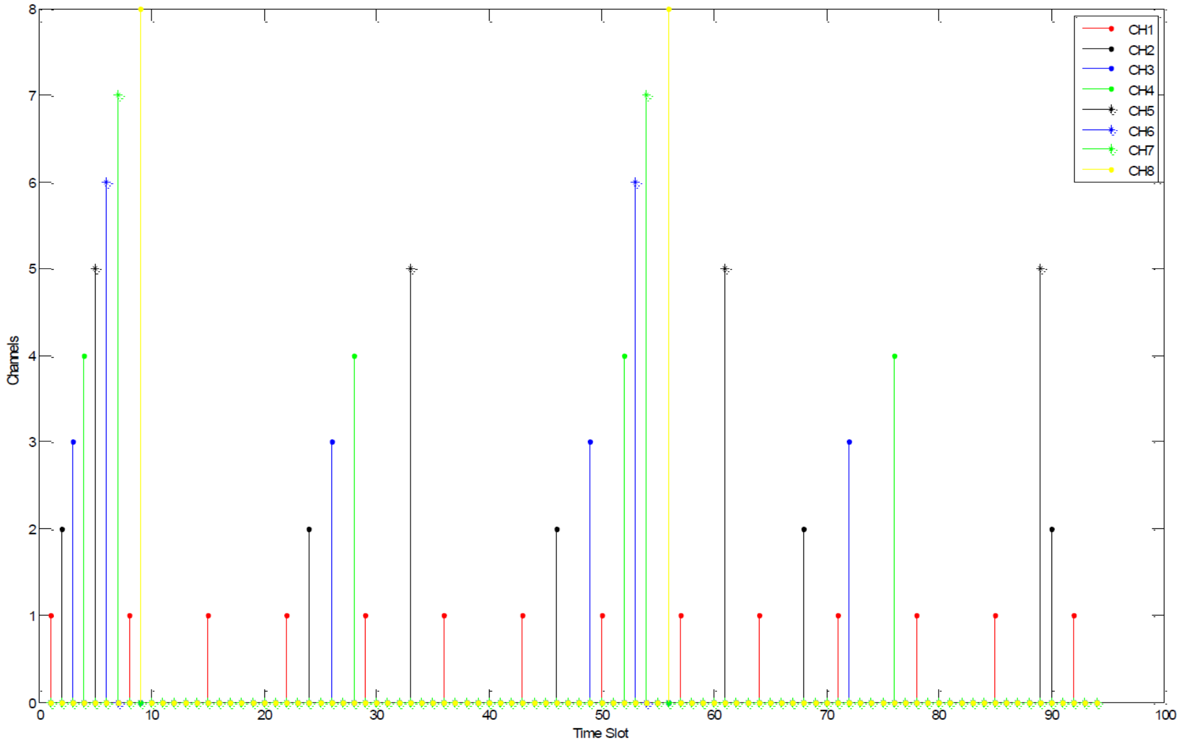


Figure 6. Eight channels with a varying sampling rate

Schedulability Analysis

Various case studies are considered for verification. One is presented in this section. Now let's consider another case study where the proposed SchoC can schedule a given task and Round-Robin technique cannot get a solution for the same task. Let the maximum sampling rate F_s of the given ADC be 100 MS/s. The time period is 10 nsec. Three analog signals are applied to the SchoC, 25 MHz, 10 MHz, and 2 MHz respectively. In addition, assume that no channel has priority. Applying the Nyquist's law, the following must be satisfied: $F_{s1} \geq 50$ MS/s, $F_{s2} \geq 20$ MS/s, $F_{s3} \geq 4$ MS/s; where F_{si} is sampling frequency for channel (i). In other words, $T_{s1} \leq 20$ nsec, $T_{s2} \leq 50$ nsec, $T_{s3} \leq 250$ nsec; where T_{si} is sampling time period for channel (i). If Round-robin sampling technique is applied, these three signals cannot be sampled using the available ADC. Applying the proposed scheduling technique, these analog signals can be optimally scheduled using the available ADC. As shown in Figure 7, the optimal scheduling is: $T_{s1} = 20$ nsec, $T_{s2} = 40$ nsec, $T_{s3} = 240$ nsec ($F_{s1} = 50$ MS/s, $F_{s2} = 25$ MS/s, $F_{s3} = 4.166$ MS/s). The tree constraints, mentioned in the optimization problem, are satisfied.

In Figure 7, the horizontal axis represents the time slots where the period between each consecutive time slots is the (T_s) which is equal to 10 nsec in this example. The vertical axis represents the channel being sampled at a certain time slot. As shown in the figure, three channels are mentioned in the vertical axis. At time slots (1, 3, 5...), channel (1) is sampled.

At time slots (2, 6, 10,...), channel (2) is sampled. At time slots (4, 28,...), channel (3) is sampled.

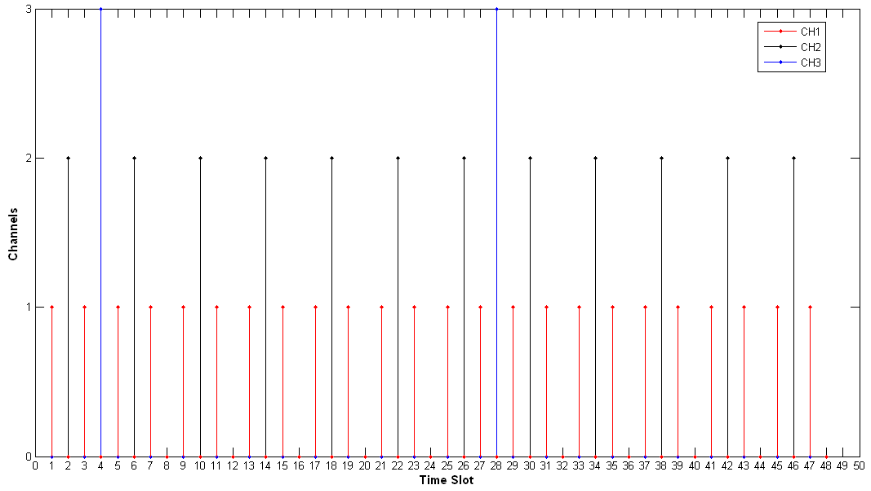


Figure 7. Round-Robin cannot schedule the given signals but the proposed SchoC can

Synthesis Analysis of the Proposed SchoC

From the Altera developing tools and the synthesis report, it is found that the prototype of the proposed SchoC only uses 11,779 logic elements (35% of the available logic elements on the FPGA), 2770 registers, 244 pins (51% of the available pins), 75,904 memory bits (16% of the available memory bits), and 4 embedded multiplier elements (6% of the available ones). It can sample up to 32 channels using a single ADC.

Dynamic SchoC

In this section, the testing of the dynamic SchoC is desvribed to verify that it can dynamically detect any change in the input channels’ frequencies. Moreover, it can adapt to the frequency change and adjust the sampling rate for each channel/signal.

Three signals are considered here: 25 MHz, 15 MHz, and 5 MHz. A single ADC (100 MSPS) is used to acquire the three signals. In the FFT calculations, 512 points are used. As shown in Figure 8, (source_real, source_imag) are the real part and the imaginary part of each output point, respectively. The signal (py) is the power spectral density (PSD) of the first 255 points. The signal (s_pyy_max) is the highest PSD and (s_pyy_max_index) is its index. In other words, (s_pyy_max_index) is the location (fci) of the greatest frequency component. In Figure 8, F_s1 means the sampling frequency for channel 1, F_s2 means the sampling frequency for channel 2, and F_s3 means the sampling frequency for channel 3. T_s1 means the sampling period for channel 1, T_s2 means the sampling period for channel 2, and T_s3 means the sampling period for channel 3. As shown, the dynamic fully hardware SchoC detects each channel’s frequency and determines the optimal sampling rate for each channel

(F_{s1}=50 MSPS, F_{s2}=33 MSPS, and F_{s3}=10 MSPS). The summation of all sampling rates is 100MSPS. It has to be less than or equals the given ADC's sampling rate (100 MSPS). Therefore, the single ADC is fully utilized in this case.

At a certain point, channel 1's input signal is changed from 25 MHz to 16 MHz. As shown in the figure, Fs_1 (the sampling rate for channel 1) is changed accordingly from 50 MSPS to 33 MSPS and the T_s1 (the sampling period of channel 1) is changed from 20 nsec to 30 nsec.

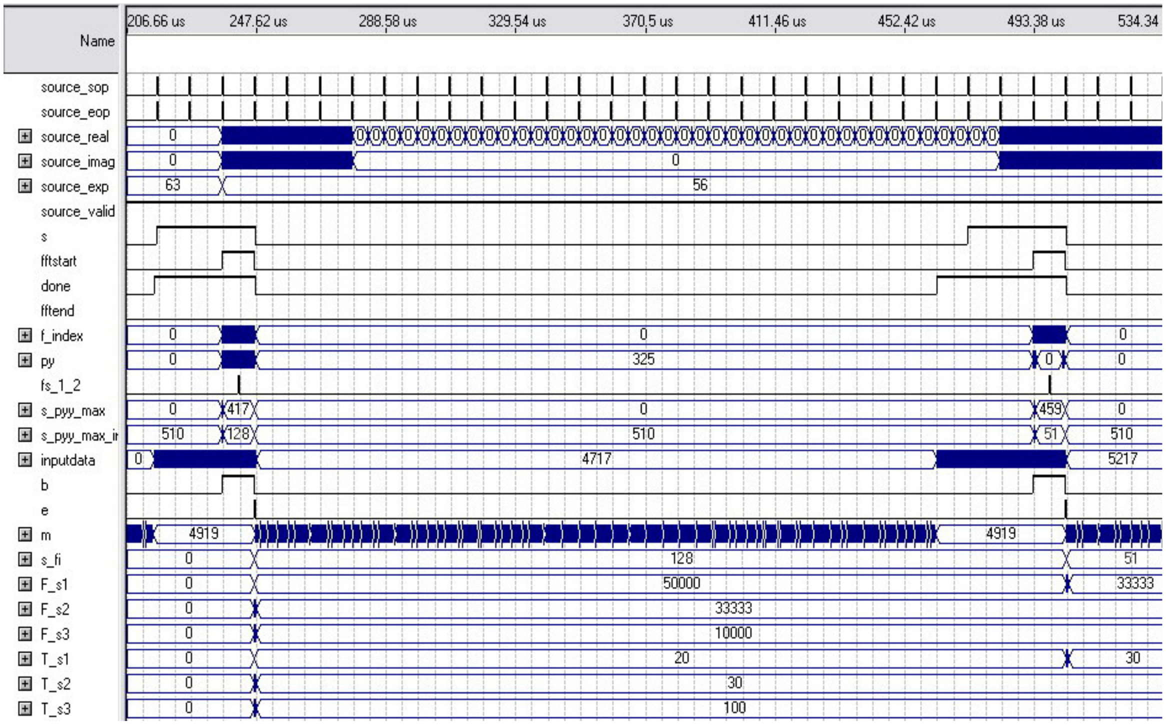


Figure 8. Dynamic SchoC (case study I)

As a second case study, three signals are considered as well. As shown in Figure 9, signals 2 and 3 are changed. Therefore, sampling frequencies for both channels are changed accordingly.

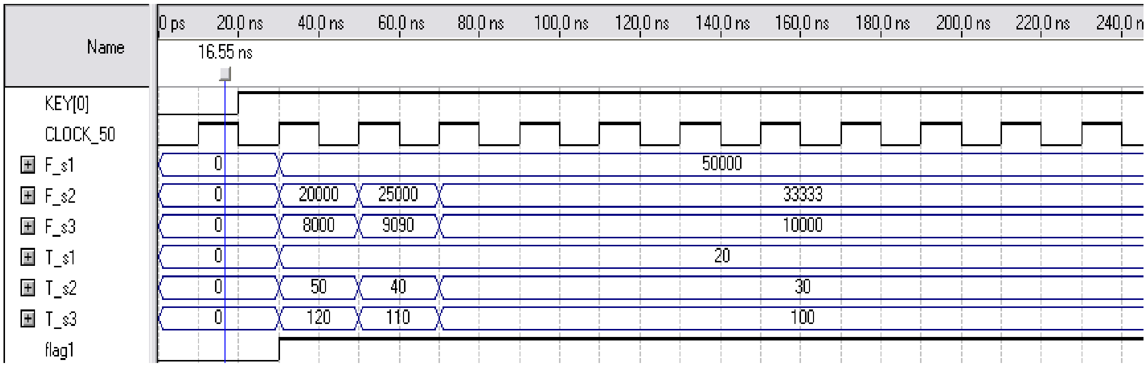


Figure 9. Dynamic SchoC (case study II)

Cost Comparison

Nallatech [11] does utilize the FPGA in its dual 3 Gsps ADC board (i.e. all main design modules are performed using FPGA), but they use a dedicated ADC per channel which in turn, increases the power consumption, cost and the circuit size. The cost of the Nallatech standalone BenADC-3G is $22,000 [11]. It has only dual channels and is not scalable. It also lacks the adaptive/reconfigurable optimized sampling technique. Moreover, according to Venture Development Corporation [12], a study regarding the global and product shifts in the DAQ market was made. They expected that the total worldwide shipments of data acquisition products will be greatly increased. In particular, a 22% increase was projected to happen in 2014, to reach $321.5 million in the USA alone.

Conclusion

In various DAQ applications, a fixed sampling rate is not the optimal solution. The dynamic scheduler-on-chip (SchoC) achieves the optimal solution for large number of channels. It also reduces memory requirements. Frequency detection and monitoring capabilities are integrated and implemented in the SchoC. In various applications, a fixed sampling rate is not the optimal solution. The SchoC achieves the optimal solution for large number of channels. It also reduces total power consumption and memory requirements. The research provided a design, implementation, and verification of a scheduler that takes full advantage of the capabilities of the FPGA as well as using a single, multiplexed ADC for multi-channel analog signal acquisition and processing. It also detects and monitors their frequencies. Any change in input channel frequency is reported via the frequency detection and/or monitoring (FREDM) system. If the input signals have different frequency bandwidths, then the SchoC is required to perform adaptive sampling, instead of using the highest frequency as a fixed sampling rate for all channels. Oversampling low frequency spectrum channels leads to unnecessary data, which in turn requires extra storage capabilities and more power consumption.

Different case studies are presented in this research. As a result, the fully hardware dynamic SchoC can schedule given tasks that are not schedulable via the Round-Robin technique. It also reduces the amount of data being acquired by up to 59%, which in turn decreases memory requirements. Moreover, the implementation of the SchoC as a hardware component on a chip achieves fast data acquisition, processing and storing system, as well as maintains an accurate signal reconstruction in terms of the frequency. The performance evaluation shows that the speed of the proposed SchoC is 24% faster than a comparable software-based scheduler. It reaches 15% better (smaller) RMS(e) than the NI software-based DAQ. Different case studies show that the proposed dynamic SchoC can detect any frequency change in the input signals. It also can adapt to the change and determines new optimal sampling rates for each channel.

References

[1] Joseph, L., & James, H. (2004). *Handbook of Scheduling: Algorithms,Models,and Performance Analysis*. Boca Raton, FL: CRC Press.

[2] Brucker, P. (2007). *Scheduling Algorithms*. (5th ed.). New York: Springer.

[3] Yurish, S. (2009). Intelligent Multichannel Data Acquisition Systems Based on Intelligent Universal Sensors and Transducers Interfacing Circuits. *IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems*, 38-43.

[4] Thomass, J., & Tang, F. (1996). A Proposed Bridge to Soil Moisture Estimation Using Current and Future Operational Satellite Radars. *Geoscience and Remote Sensing Symposium*.

[5] Alex, H., Robert, M., & Fred, R. (2012). A Multichannel Continuously Selectable Multifrequency Electrical Impedance Spectroscopy Measurement System. *IEEE Transactions on Biomedical Engineering*, *25*(3), 49-57.

[6] Avramova, Z., Wittevrongel, S., Bruneel, H., & Vleeschauwer, D. (2009). Dimensioning of a Multi-Rate Network Transporting Variable Bit Rate TV Channels. *IEEE International Conference on Communications*, 1-6.

[7] Rieger, R., & Taylor, J. (2009). An Adaptive Sampling System for Sensor Nodes in Body Area Networks. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *17*(2), 183-189.

[8] Abdallah, M., & Elkeelany, O. (2011, August). A Low Cost Stand-Alone Multi-Channel Data Acquisition Monitoring and Archival System with On-Chip Signal Pre-Processing. *IEEE Transaction on Instrumentation and Measurement*, *60*(8), 2813–2827.

[9] National Instruments. (2013). *PCI-6115*. Retrieved from http://sine.ni.com/nips/cds/view/p/lang/en/nid/11886

[10] Texas Instruments. (n.d.) *ADC12D040*. Retrieved from http://www.ti.com/product/adc12d040

[11] Nallatech. (2010). *Virtex-4, Dual 3 GSPS ADC*. Retrieved from http://www.nallatech.com

[12] Taylor, J. (2008). *Global and Product Shifts in the Data Acquisition Market*. Boston, MA: Venture Development Corporation Press.

Biography

M. Abdallah got his Ph.D. in Electrical Engineering and works as an assistant professor in SUNYIT. His research area is reconfigurable computing and embedded systems. He works as a reviewer for many journals. He has published more than 40 conference papers and journal articles.