Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Hardware Reconfigurable Secure Data Acquisition System of a Shared ADC in Heterogenous Multi-channel Inputs with Automatic Sampling Frequency Configurations

oalkelany Alkelany

oalkelany@horus.edu.eg

Horus University-Egypt

Article

Keywords: Real-time concurrent process scheduling, Time multiplexing of analog to digital converters, Computer aided design, Reconfigurable devices, Hardware software codesign, Concealing secret key

Posted Date: August 4th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10301685/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Version of Record: A version of this preprint was published at Scientific Reports on November 9th, 2026. See the published version at https://doi.org/10.1038/s41598-026-73573-w.

Hardware Reconfigurable Secure Data Acquisition System of a Shared ADC in Heterogenous Multi-channel Inputs with Automatic Sampling Frequency Configurations

Omer Alkelany, *Senior Member, IEEE, oalkelany@horus.edu.eg*

***Abstract*—** Multi-channel Data Acquisition (DAQ) Systems can use a single multiplexed analog to digital converter (ADC) to reduce system cost. But this ADC must be configured to proper sampling rate for each channel. In this paper, a novel algorithm is proposed to automatically generate ADC configuration, and improve the ADC's utilization and utility, reduce the number of redundant bits, and reduce cost. This proposed algorithm is mathematically formulated, and numerical examples are given. The research methodology involves using FPGA board with analog to digital and digital to analog conversion card, and synthetic wave form is generated and captured for verification. In this phase, a single channel is investigated, as the investigation of multi-channel DAQ requires the integration and scheduling of the ADC with an external analog multiplexer. In prior work, a custom computer board is designed to implement the HTMS algorithm for 32 analog channels. To effectively demonstrate the previously proposed HTMS superiority against other approaches, new performance metrics were defined, namely the ADC's utilization, utility and efficiency using two different heuristic analog input sets. These metrics show that the HTMS-based DAQ outperforms relevant systems as it reaches 100% of ADC's utilization and utility, while other systems have 75% utilization, and 6.25% utility. By devising a unified figure of merit for multiple DAQ systems, it is shown that the proposed system is higher by 218% compared to the baseline multi-channel DAQ system. Furthermore, Advanced Encryption Standard (AES) has been extensively investigated for cryptographic applications. A given pipelined 54-cycle AES-256-bit key core is considered for enhancement. With a single cycle design approach, the given core is successfully implemented in FPGA device using only 14-cycles / round, and generating a maximum encryption throughput of 12.8 Gb/s, with an FPGA utilization of only 9% Cyclone V SX SoC 5CSXFC6D6F31C6N, with speed up factor of 7 times is reported, compared to recent results, operating at a maximum clock frequency of 100 MHz. The proposed system will have a broader impact, particularly in medical diagnosis and prognosis, where multiple sensors of heterogenous types are used simultaneously, and the collected data must be secured to protect subjects' privacy.

Keywords- Real-time concurrent process scheduling, Time multiplexing of analog to digital converters, Computer aided design, Reconfigurable devices, Hardware software codesign, Concealing secret key.

I. INTRODUCTION

The Nyquist-Shannon sampling theorem stipulates that any bandlimited continuous time analog signal can be accurately reconstructed by acquiring samples at a sampling rate at least twice the maximum frequency component of its frequency domain [1] [2]. Therefore, many research directions have been investigated for data acquisition systems such as the nonuniform sampling [3], [4], [5] and multi-channel sampling [6], [7], [8], [9], [10], [11]. For multi-channel data acquisition systems one approach is to use a *dedicated* Analog to Digital Converter (ADC) per channel such as in [6], [8], [11], [12], [13], [14], [15], [16]. Also, it is possible to use code division multiplexing approach to separate multi-channels by performing digital phase correction, or code division multiplexing [17], [18]. Another approach is to use a multi-channel ADC device, which is internally scheduled uniformly [7], [10]. Some of the new research directions are on non-uniform sampling and signal reconstruction in the presence of packet loss [3], [4], [5], but were focused only on single channel input. Also, some researchers performed simulations only and have not performed real time realizations of their techniques [5].

A common approach for multi-channel data acquisition is to use a dedicated ADC per input channel. But some ADCs are very expensive due to their high requirements. For example, the Texas Instruments ADC34RF55IRTD, which is a dual channel 14-bit sampling rate of 3G sample/s is about $1000. If the application requires 32 channels, for example of this ADC, then 16 of this device will be needed. Another example is the 12-bit, single channel, pipelined 10.25 G sample/s, the JESD204B ADC from Analog Devices, which costs more than $21K. So, this high ADC cost will increase the overall system cost, which affects a variety of applications (i.e., tele-communication, industrial control systems, wearable computers, medical devices, etc.) [19], [20], [21], [22], [23], [24]. High speed ADCs are also needed in high processing capabilities for artificial intelligence applications when various high frequency inputs are to be captured simultaneously to allow time-frequency correlations. So, a fundamental challenge is: how to enable these applications effectively.

An effective technique for multi-channel data acquisition system is to use a single, shared, time-multiplexed ADC, such as shown in Figure 1. To maintain analog signal quality, known issues such as settling time delay, memory effects, and inter-channel cross-talk are addressed with signal preconditioning and oversampling rates to avoid possible transient errors due to channel switching [25], [26], [27]. The time-multiplexed single shared ADC technique is already implemented by industry, such as in the (TI ADS-7953) device. In this device, the user is given the ability to program the schedule of channel selection in the configuration process of the ADC. Therefore, this type of ADC programming can be considered as real-time system with concurrent process scheduling problem. That is because the multi-channel analog inputs must be filtered from noise, acquired, quantized, and stored all with real-time processing requirements. A simple, but not always effective multi-channel selector algorithm is to consider all channels *similar*, and connect them to the input of the analog multiplexer with *equal time periods*, such as in the Round Robin (RR) for equal priority scheduling.

The time-multiplexed ADC can be shared and scheduled to sample multiple inputs using an analog multiplexer. Research

work in [25] used pre-set ADC timing configurations to acquire various types of biomedical signals, such as ECG/EEG etc.

In real-time computer processing, a schedule is infeasible if a periodic task misses its processing time slot, also referred to as it misses its specified deadline. So, in multi-channel time-multiplexed ADC, a pattern that allocates the single ADC to acquire two or more channels *at the same time*, will result in data corruption problem. The author calls this problem Conflicting Time-Samples (CTS) problem demonstrated in Figure 2 [25], [26]. Such problem must be avoided to maintain integrity of the acquired data. When the multi-channel analog inputs have diverse maximum frequency harmonics, the use of a simple equal sampling time for all channels is ineffective. So, a solution is needed to automatically search for a schedule, and apply it to the schedule of the ADC. This is addressed in this research.

The key challenges of multi-channel DAQs are:

- Single channel Data Acquisition Systems (DAQs) require the knowledge of the frequency harmonics of its input signal, so that the maximum frequency harmonic of the input is multiplied by the Nyquist rate which is the minimum required for accurate reconstruction and an oversampling factor to improve quality of acquisition. The automatic detection of these frequency harmonics can be done via Fast Fourier Transform. A challenge exists when the maximum frequency harmonic changes in real-time. To remedy that, the process of *learning the maximum frequency harmonic has to be done periodically to ensure accurate reconstruction of the input signal*. However, this is not addressed in this research article.
- Available multi-channel DAQs e.g., NI USB 6453, ADS 7953, use uniform scheduling. A uniform scheduling pattern for heterogenous inputs sets the sampling frequency according to the highest maximum frequency harmonic, across all channels. If the maximum frequency harmonics are diverse (i.e., widely separated), then uniform scheduling may result in *excessive redundant data*. Some ADCs allow the user to program the schedule of the channel selection such as the ADS 7953.
- Multi-channel sample scheduling techniques, such as WRR [28], RM [29], may suffer from Conflict Time Sampling (CTS) problem where the sampling schedules two channels at the same time slot. Since CTS problem results in data corruption [25], the proposed algorithm in this research *avoids CTS problem automatically.*
- Some of the recent research work used reconfigurable hardware to implement DAQs such as in [6], [7], [25], [26], [30] for managing systems' commands and its timings. Other recent researchers used multi-channel DAQs for biomedical signals such as in [31] [32] [33] [34]. Preliminary research work in this article also used FPGAs. However, *none of these research works have addressed the possibility of integrating data privacy protection core on the same FPGA device*.
- Some of the Performance measures (e.g., ADC utilization, ADC utility, generation of redundant bits, cost, etc.), power consumption, and cost often conflict with each other. *Finding an effective solution is the main challenge addressed in this research frequency harmonics of the input channels are assumed to be given to the scheduling algorithm.*

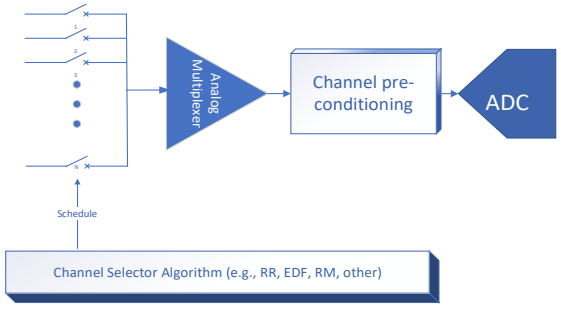

Fig. 1. Multi-channel data acquisition for N channels, and a single analog multiplexer, shared channel preconditioning and a shared analog to digital converter. Appropriate channel selector algorithm is needed.

Research objectives:

The goal of this research is to address the previous challenges and develop an effective algorithm for a single, shared ADC to acquire heterogenous input channels via a time-multiplexing approach at real-time, using heterogenous sampling, and avoid Conflicting Time-Samples automatically.

The research objectives are to:

- Demonstrate how our previously proposed HTMS sampling selector algorithm [35] [36] [37] which automatically generate a minimum schedule length to acquire the multi-channel input signals, of heterogenous frequency characteristics can be implemented on the software side of a FPGA chip.
- Demonstrate how the proposed algorithm successfully avoids Conflicting Time-Samples (CTS) problem, by inserting necessary sampling shifts in the proposed algorithm.
- Develop new performance evaluation metrics and show how the HTMS reduces the number of redundant bits after the conversion process for multi-channel DAQs,
- Improve the utilization of a single ADC for multi-channel data acquisition systems,
- Reduce the power consumption and cost for a DAQ system, by using a single ADC of high enough sampling frequency for multi-channel analog inputs,
- Integrate a fully-pipelined enhanced hardware encryption core, with the proposed HTMS-DAQ system, to offer data protection for real-time acquired data on the same FPGA chip.

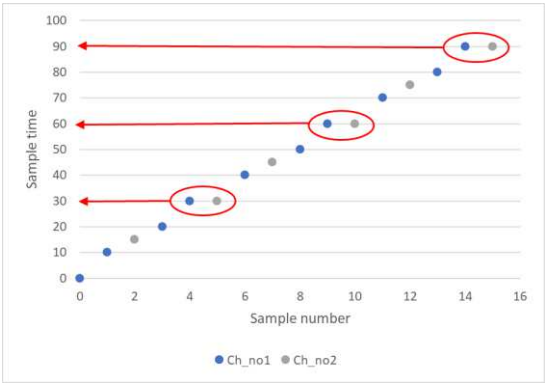

Fig. 2. A demonstration of the Conflicting Time-Samples for two channels at sample times 30, 60, 90 ms.

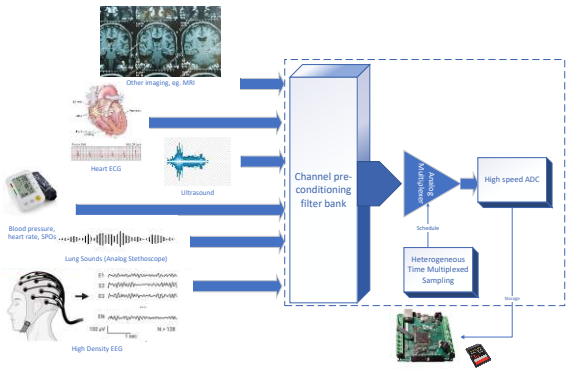

Fig. 3. A demonstration of an example system with Heterogenous input channels and single ADC.

Research Key Performance Indicators (KPI)s:

A previously developed Heterogenous Time-multiplexing Algorithm (HTMS) [35] [36] [37] that avoids conflicting time samples at real-time, is presented for completeness of the topic. Then:

- ✓ KPI 1, Satisfactory design of an integrated hardware/software system that has an integrated ADC scheduling side on the software side, and encryption core and other required processing capabilities on a single FPGA chip,
- ✓ KPI 2, Achievement of a figure of merit for the proposed system not less than 2 compared to nearest known relevant system.
- ✓ KPI 3, A final performance indicator is the successful encryption of acquired data at an accumulative data rate not less than 12 Gb/s, or 1 G sample/s for 12 b/sample ADCs.

Research methods

Firstly, and for completeness of the topic, the Conflicting Time-Samples (CTS) problem [25] is demonstrated, and its avoidance is shown briefly. Then the previously developed Heterogeneous Time-multiplexing Algorithm (HTMS) [35] [36] [37] is represented mathematically with a numerical example to demonstrate its merits (KPI1). HTMS combines the features of non-uniform scheduling, LCM based repeatable sampling window, and sample offset assignments to avoid timing conflicts. Then the evaluation of merits is done using newly developed metrics, and a custom defined FoM (KPI2). Finally, the proposed HTMS-DAQ system is integrated with an enhanced AES-256 pipelined encryption core on the same FPGA device (KPI 3).

Limitations of Existing Multi-channel DAQs

In multi-channel Data Acquisition Systems (DAQs) with a single ADC, a simple possible scheduling solution is finding a constant largest possible sampling rate for all channels, regardless of the redundant data, high power consumption, and expensive component requirements. But this solution is not always effective. Weighted Round Robin (WRR) scheduling [36], which can be used to allocate multiple ADCs based on their capacities. Then a cyclic approach to collect resulting data will be performed based on the sampling rates of each ADC (i.e., the weights). Although useful, but ineffective to reduce cost, size, and power consumption. However, other algorithms such as EDF, or RM [29] can be used for a single ADC. But since they are designed to give priority based on sampling rate, they don't have a mechanism to avoid CTS problem.

Homogenous scheduling of diverse input channels to the same ADC may suffer the following limitations

- High Redundant bits
- Limited ADC Utility
- Limited Utilization

Also, to the best knowledge of the author, there was no privacy protection core integrated with the scheduling module on the same chip of the system before. This integration, which is the motive of this research, allows the real-time encryption of the acquired data from DAQ system, on a single device, reducing system cost, and improving its efficiency.

II. Proposed HTMS Technique

Figure 2 demonstrates an example of heterogenous acquisition of diverse analog channels in the medical field. Various imaging and sensor signals are analog in nature. The analog input characteristics are obviously very diverse, and the concurrent acquisition of these diverse inputs is useful for medical prognosis and diagnosis. As the costs of the ADC can be very high, a shared single ADC can be used, if scheduled properly. The generated schedule should not have timing conflicts for the ADC, to maintain integrity. The development of an algorithm to automatically find such schedule is herewith called (HTMS).

The CTS problem [25] occurs when the scheduling algorithm assigns two samples to be acquired at the same time. Using a single ADC this results in data corruption, and must be avoided. The HTMS is designed to avoid CTS automatically, by ensuring that samples of all given channels of different sampling times are separated in time. Figure 2 shows an example of CTS for 2 channels.

A. HTMS based DAQ system goal and objectives

The main goal of the proposed HTMS is to automatically generate a schedule for a single ADC to acquire heterogenous input channels via a time-multiplexing scheduling approach. The objectives of the HTMS are to:

- Automatically generate a minimum schedule length to acquire the input signals.
- Automatically avoid Conflicting Time-Samples (CTS) problem, by inserting sampling shifts.
- Reduce the amount redundant data bits for the accurate reconstruction of analog inputs,
- Efficiently maximize a single ADC utilization, and its utility,
- Reduce the cost for a multi-channel Data Acquisition (DAQ) system, by only using a single ADC of high enough sampling frequency needed for analog input specifications.

Compared to traditional homogenous sampling, HTMS based DAQ will reduce the need for excessive storage of output data based on same frequency of sampling.

To demonstrate this, we provide the following definitions.

Definitions:

Let N be the total number of channels, f_{MAX} be the given maximum sampling frequency the ADC (Hz), which might also be given as samples/s (Sa/s).

Let *Fn* be the working set of suitable sample rates for the *nth* channel of the ADC, thus $0 \leq n < N$.

Let the oversampling factor (Φ) be 5 times the Nyquist rate,

Let the utilization U of the multi-channel ADC be:

$$U=2*\Phi*N*Fm/fmax, \quad (1)$$

where Fm is the maximum frequency harmonic of the input signal, and *f*max is the maximum sampling frequency of the shared ADC.

Let the utility of the multi-channel time multiplexed ADC defined by:

$$\Psi= 100* \text{ number of functional channels/N}, \quad (2)$$

The utility Ψ can be less than 100%, when some of the input channels are of higher frequency harmonics that demands higher sampling rates, and there are not enough sampling sloths for some of the existing channels.

Let the redundant data bits/s (ϑ), be the number of bits acquired for sampling an input channel with a sampling rate more than 2*Φ*N*Fm. This (ϑ) can be >0 in homogenous scheduling of single shared ADC when input signals of low frequency harmonics are present on channels of the ADC simultaneously with input signals on other channels of high frequency harmonics.

Now consider the case study of a set of 32 analog inputs, time multiplexed for acquisition using a single 1Ms/s multi-channel 12-bit ADC shown in Table I. Let the oversampling factor (Φ) be 5 times the Nyquist rate, and that each of the analog inputs has a maximum frequency harmonic of 3.125kHz. This results in a required sampling rate of 31,250 Sa/s for each of the 32 channels. Using equations (1) and (2) U, and Ψ are:

$$\frac{=100*2*5*32*3.125*10^3}{10^6} \%^{=} 100\%.$$

Ψ= 100%,

respectively in this case. The number of generated bits/s for all channels (Θ) = 12 Mb/s, because each sample is 12 bits, and (ϑ)=0, since all bits are necessary for accurate reconstruction of input signal. The efficiency can be computed using:

$$E=100*(\frac{b*fmax-\vartheta}{b*fmax})\%, \quad (3)$$

where b is the number of bits/sample (b/Sa). In this ideal case, E=100%.

The proposed HTMS will automatically generate a schedule of acquisition for channel 0 at a rate equal to $500*10^3$ samples /s, which is the maximum sampling frequency *fmax*, and less sampling frequencies to all the other channels (1 to 31) ranging from $250*10^3$ samples /s, to 500 samples/s for channel numbers 12 to 31. Since the summation of required samples/s does not supersede fmax, the schedule is feasible.

Using equations (1) and (3), the values of U and E for the HTMS based scheduling can be recomputed:

The minimum repeatable window size is 10000 samples, of duration 10 ms, and channels will be sampled at a rate of:

5000, 2500, 1250, 400,400,50,50,50 (7 times), 5, 5, 5 (15 times) ,1,1,1,1,1 times in every 10 ms.

The summation of the number of samples in the window is also 10000, so U=100%, E=100%, and using equation (2) the utility (Ψ) is also 100%, as all 32 channels of the system are operational.

However, the conventional uniform scheduling algorithm will fail for one of the following reasons:

1) If all 32 input channels specified in Table I, are uniformly sampled with a uniform sampling rate equal of 500,000 samples/s, a faster ADC will be needed at a speed 16 M samples/s. Even if the ADC is changed, a large number of redundant bits will be generated per second for all channels of lower frequency harmonics, as shown in Table II. Table II shows the total redundant bits $\vartheta=180.024*10^6$, where the maximum number of bits/s is $12*16*10^6=192$Mb/s. Using equation (3) $E=100*(192*10^6-180*10^6)/192*10^6$ Therefore, the efficiency E will be only 6.25%, but the utility Ψ is kept at 100%, since all 32 channels are operational. The ADC's utility (Ψ) in this case is still 100%.
2) If the same ADC of 1 Ms/s is used instead, uniformly sampling all input channels at 500 ks/s, then only channels

TABLE I:
MULT-CHANNEL INPUTS WITH FREQUENCY DEVIATIONS

n: channel number	Fm: Maximum Frequency Harmonic (Hz)
0	45000
1	20000
2	10000
3-4	3500
5-11	500
12-26	50
27-31	10

TABLE II:
REDUNDANT BITS FOR UNIFORM SAMPLING OF CASE STUDY, USING 12B RESOLUTION

n: channel number	Fm: Maximum Frequency Harmonic (Hz)	Chosen sampling rate # (sample/s)	Redundant bits /s (500000-chosen sample/s) *12 from Table I
0	45000	500000	
1	20000	250000	250000*12
2	10000	125000	375000*12
3-4	3500	40000	460000*12*2
5-11	500	5000	495000*12*7
12-26	50	500	499500*12*15
27-31	10	100	499900*12*5
		Total	180.024 Mb/s

The chosen sampling rate is ≥ 2*Φ* Fm, for even distribution

0 and 1 can be operational, keeping the utilization of the multi-channel DAQ based on a single ADC at 100%, but generating redundant data bits of ϑ=3 *10^6/s, where the maximum number of bits/s is 12Mb/s. So, the efficiency is: E= 100*(12-3)*10^6/12*10^6=75%. But this choice reduces the ADC's utility (Ψ) to 6.25%, since only two channels of the possible 32 are operational.

Table V, in section 4 summarizes these findings.

Therefore, it can be seen that the proposed HTMS can improve multi-channel DAQ systems, gives a best feasible schedule for a single time-multiplexed ADC and keeping its utilization, utility, and efficiency as high as possible.

The reduction of power consumption and cost can be shown as follows. An expensive single ADC that is capable of sampling all heterogenous inputs at the maximum sampling frequency needed by the highest channel requirement is no longer needed. A fast enough single ADC can be used, and thus reducing the cost and power required. Using the examples illustrated before, the cost and power consumption of the 12-bit 16 Ms/s ADC is higher than the cost of a 12-bit 1Ms/s for obvious reasons.

- Feasibility: if the sum of all sampling periods is less than or equal to the ADC sampling period, then the HTMS algorithm will dynamically produce a repeatable pattern for heterogenous input acquisition.
- Limitation: the summation of sampling periods must be less than or equal to the minimum sampling period of the given ADC.
- Observation: the generated sampling periods may not be absolutely optimal. HTMS is based on integer optimization, and due to rounding errors, some values are rounded up to find the solution.
- Assumption: the frequency characterization of all input signals is static during the learning process of the HTMS. This means that input channels are assumed to hold their characteristics, (maximum frequency component at real time), after the learning phase. This is always a given assumption, otherwise the generated output cannot be used to reproduce the inputs.

B. Mathematical Representation of the Proposed Novel HTMS Algorithm

The objective of the innovative HTMS is to find the working set of time-shift in sampling time so that all heterogenous channels are acquired without the Simultaneous-Time Sampling (CTS) problem. Since the factors are chosen to divide f_{MAX} evenly, they will also divide evenly all multiples of f_{MAX}. Therefore, pattern of length l is sought by the HTMS, so that it can be repeated for by DAQ during data acquisition effectively. Notably, the pattern length l may not be integer. The pattern length l, however, has to be an integer multiple of $1/f_{MAX}$.

Let f be the set of multi 4-element vectors containing the required channel information as follows:

$$f \equiv \left\{ f_0 = \begin{bmatrix} ch_no_0 \\ \rho_0 \\ \sigma_0 \\ \omega_0 \end{bmatrix}, f_1 = \begin{bmatrix} ch_no_1 \\ \rho_1 \\ \sigma_1 \\ \omega_1 \end{bmatrix}, \dots, f_n = \begin{bmatrix} ch_{no_n} \\ \rho_n \\ \sigma_n \\ \omega_n \end{bmatrix}, \dots f_{N-1} = \begin{bmatrix} ch_no_{N-1} \\ \rho_0 \\ \sigma_{N-1} \\ \omega_{N-1} \end{bmatrix} \right\}, \quad (4)$$

where:

$0 \le n < N$

ch_no_n be the channel number that the ADC must acquire, in the pattern number n

ρ_n is the minimum sampling rate in Hz for the channel ch_no_n

σ_n be selected sample rate for ch_no_i

ω_n be the offset (in samples) from the start of the pattern for channel ch_no_n

For the HTMS algorithm, f is sorted in descending order by ρ, (i.e., ρ_0 is the largest). The first vector in f, i.e. f_0 represents the highest priority of the list of channels to be scheduled, and the schedule ends with the least priority or with f_{N-1}. For each channel, a sample rate σ_n is selected to be the smallest solution in the working set of Fn, such that it is greater than or equal to ρ_n. Therefore, the HTMS, sets σ_n, to be the closest solution from the working set of frequencies from the F_n, such that $\sigma_n \ge \rho_n$.

F_n, defined before, is expressed as:

$$F_n = \begin{cases} factors\,(f_{MAX}),\ n = 0 \\ factors(\sigma_{n-1}) \cap F_{n-1},\ n > 0 \end{cases}, \quad (5)$$

The feasibility condition for the HTMS is based on the following summation:

$$\sum_{n=0}^{N-1} \sigma_n \le f_{MAX}\ , \quad (6)$$

This summation must not be greater than f_{MAX}:

The HTMS generated pattern of length l (in samples) is found using Least Common Multiple (LCM) function. The largest ratio between f_{Max} and any sampling rate is found. Then the LCM of this largest ratio and the ratio between f_{Max} and any sample rate for other channels can be found.

Mathematically, this is done by finding:

$$\sigma_{mim} = minimum(\sigma_i) \quad \forall_i$$

$$l = LCM\left(\frac{f_{MAX}}{\sigma_{min}}, \frac{f_{MAX}}{\sigma_i}\right); \quad \forall_i \quad (7)$$

To determine the shift ω_i, the process is as follows:

Tn = $1/\sigma_n$; The sample time of channel n,

Tmin=$1/f_{Max}$, The ADC minimum sampling time,

where

ω_0=0 (i.e., no shift) because the pattern starts with channel of the maximum sampling frequency.

ω_1=1, because the second sampling slot will always be available as long as more than one channel is to be acquired. For all other channels, the difference d between the shifts of channel n and the ith previously chosen channel d=(ω_n-ω_i) can be computed. However, this d must be tested to be not an integer multiple of the all previously scheduled channels, to avoid the CTS problem. Since the HTMS is designed to find the smallest time shifts ω_n such that no CTS occurs, the HTMS is set to find the minimum ω_n such that:

$$\frac{(\omega_n - \omega_i)T_{min}}{T_i} = \Delta, \quad \forall_i, \quad (8)$$

where Δ is not an integer.

C. Numerical Example for the HTMS working

Let N=3 channels, f_{MAX}=20 Hz, the minimum sampling rates ρ_i for the 3 channels be:

4,2,10 Hz for channel numbers 0,1,2.

So, the initial f (sorted in descending order by ρ_i) is computed using (4) as:

$$f \equiv \left\{ f_0 = \begin{bmatrix} 2 \\ 10 \\ \sigma_0 \\ \omega_0 \end{bmatrix}, f_1 = \begin{bmatrix} 0 \\ 4 \\ \sigma_1 \\ \omega_1 \end{bmatrix}, f_2 = \begin{bmatrix} 1 \\ 2 \\ \sigma_{N-1} \\ \omega_{N-1} \end{bmatrix} \right\},$$

For the first (i.e., 0th channel), F_0, is the integer factors of f_{MAX} :

F_0={1,2,4,5,10,20}

To select the sample rate σ_0, the HTMS is set to find the smallest working factor from F_0, $\geq \rho_0$. So, it finds: σ_0=10.

To select the sample rate σ_1, the HTMS is set to compute F_1, the set of working factors in F_0 that are also working factors of σ_0. So, it sets:

F1={1,2,5,10}

Consequently, σ_1, the least factor from F_1, $\geq \rho_1$.

Since ρ_1=4, the HTMS sets σ_1=5, which is $\geq \rho_1$

Similarly, the HTMS will populate the elements of F_2, all the working factors in F_1 that are also factors of σ_1. So:

F_2={1,5}.

Then, the HTMS is set to σ_2 the smallest working factor in F_2 that is $\geq \rho_2$. Since ρ_2=2, the HTMS sets:

σ_2=5, which is $\geq \rho_2$.

Consequently, *f becomes*:

$$f \equiv \left\{ f_0 = \begin{bmatrix} 2 \\ 10 \\ 10 \\ \omega_0 \end{bmatrix}, f_1 = \begin{bmatrix} 0 \\ 4 \\ 5 \\ \omega_1 \end{bmatrix}, f_2 = \begin{bmatrix} 1 \\ 2 \\ 5 \\ \omega_{N-1} \end{bmatrix} \right\}$$

To check the feasibility, the feasibility requirement (6), it can be found that: 10+5+5≤ 20. Since the solution is feasible, the HTMS is also set to compute the shift values ω_n, to avoid CTS problem. As illustrated before, the following can be observed:

ω_0= because the first channel to be acquired will always be at the beginning of the pattern.

$\omega_1 = 1$ because the second sampling slot will always be available for two or more heterogenous input channels

To find ω_2, a choice of the minimum value in all positive integers must be done such that:

$$\frac{(\omega_2 - \omega_i)T_{min}}{T_i} = \Delta, \quad \forall_i,$$

$Tmin = 1/f_{MAX}$=50ms

The values of 0 and 1 are already used, so the value 2 can be considered, to compute Δ. but:

$$\Delta_i = \frac{(2-0)0.050}{0.100} = 1, i = 0$$

This value is an integer, and hence must be avoided. Then the value 3 can be considered to compute Δ_i to verify.

$$\Delta_i = \frac{(3-0)0.050}{0.100} = 1.5, \ i = 0$$

$$\Delta_i = \frac{(3-1)0.050}{0.200} = 0.5, \ i = 1,$$

Since both values of Δ_i are non-integers, the value of 3 is assigned to ω_2. The final multi-dimensional vector f is

$$f \equiv \left\{ f_0 = \begin{bmatrix} 2 \\ 10 \\ 10 \\ 0 \end{bmatrix}, f_1 = \begin{bmatrix} 0 \\ 4 \\ 5 \\ 1 \end{bmatrix}, f_2 = \begin{bmatrix} 1 \\ 2 \\ 5 \\ 3 \end{bmatrix} \right\},$$

Finally, the HTMS finds the pattern length, *l*.

This pattern length l is computed by first finding the lowest chosen frequency σ_{min}. This in turn is found by computing the minimum $\sigma_i \quad \forall_i$:

σ_{min}= min (5,5,10) = 5

Then, the HTMS is set to compute *l* by finding the LCM of lowest chosen sampling periods using:

$$l = LCM\left(\frac{f_{MAX}}{\sigma_{min}}, \frac{f_{MAX}}{\sigma_i}\right); \quad \forall_i,$$

l=LCM (20/5, 20/5, 20/10) =4

Figure 4 shows the generated schedule of the HTMS algorithm for this example. Note that three channels are numbered as 0,1,2 on the vertical axes. This schedule satisfies the input channel sampling requirements, and does not suffer CTS problem. The figure shows the pattern repeated twice.

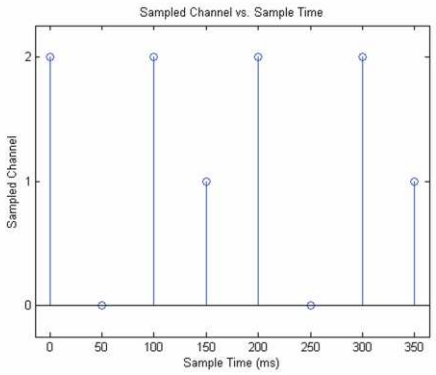

Fig. 4. HTMS generated pattern repeated twice.

Table III compares between the HTMS and other known task scheduling algorithms. As shown in the table, the HTMS is classified as a non-uniform, LCM-based, Timing offset assignment scheduling algorithm, to avoid timing conflicts.

TABLE III:
COMPARISON BETWEEN HTMS AND F RELEVANT SCHEDULING ALGORITHMS

Scheduling Algorithm	Periodic task scheduling	Uniform scheduling (Static)	Non-uniform scheduling (Dynamic)	LCM based repeatable window	Timing Offset assignment
HTMS (this)	√	**x**	√	√	√
EDF	√	**x**	√	**x**	**x**
RM	√	√	**x**	**x**	**x**
WRR	√	**x**	√	**x**	**x**

III. SECURING ACQUIRED DATA OF THE HTMS-BASED DAQ

Conventional DAQs do not integrate data security to the system. In many applications, acquired data has a privacy requirement. It is well known that the Rijndael algorithm was selected by NIST in 2001 and given the name Advanced Encryption Algorithm (AES) [38]. Two cryptographers, Joan Daemen and Vincent Rijmen, created the AES with a constant block size of 128 bits and key sizes of 128, 196, or 256 bits [39].

The AES can be implemented in a variety of modes. The most straightforward is the Electronic Code Block mode (ECB), in which each input block is encrypted separately and independently of other input blocks. In order to boost the encryption rate in pipelined AES implementations, the ECB mode is therefore highly helpful. Cipher Block Chain mode (CBC), Cipher Feedback (CFB), and Output Feedback (OFB) are additional AES modes, but cannot be used effectively in pipelined manner [40]. These alternative modes increase the output's reliance on the order of input blocks by applying cipher procedures in a chained manner. Reconfigurable hardware is known to enable designs resistant to side-channel attack, and the AES-256-bit key is known to be resistant to quantum computing attacks.

As stated earlier, AES-ECB mode can be attained with great speed since it processes several blocks at the same time. Different optimization methods were investigated for AES-ECB, including basic looping, loop unrolling, partial pipelining, partial pipelining with sub pipelining, and complete pipelining [41]. The AES algorithm is structured to perform fundamental operations on the bytes of the input block, which is arranged as a 4 x 4 matrix of 16 bytes, resulting in the 128-bit input block. The block size for AES input is constant, but the key size can be expanded.

The process of encryption necessitates that fundamental functions be applied repeatedly in cycles. The overall number of rounds relies on the key size. The input and output for every step in the round is the 4x4 matrix, which is referred to as the state. The state evolves as the actions are executed. The most basic version of AES uses a 128-bit key and necessitates 10 rounds [38]. The decryption is essentially the reverse of the encryption process, executing the cryptographic actions in the opposite sequence. As noted before, the input block size is set at 128 bits, whereas the key size may be 128, 192, or 256 bits. Every data block is arranged in a matrix structure. The AES algorithm operates in iterations, with each iteration referred to as a round. The designated number of rounds is 10, 12, and 14 for key sizes of 128, 192, and 256 bits correspondingly. The AES begins with a *KeyExpansion* process, in which a unique *RoundKey* is generated from the original secret key for every round. Additionally, prior to the first round, the input block is combined with the secret key via the XOR operation to produce the initial state. This is the last step to produce the encrypted block [39].

Numerous AES implementations focused on single-core processors within FPGA designs. This approach demonstrated encouraging encryption speeds for AES-ECB mode, including no pipelining [42] [43], folded parallel [44], and fully pipelined [41]. This approach emphasizes attaining elevated encryption/decryption speeds, utilizing different key sizes. Certain research findings excluded the possible cryptographic threats. Moreover, the combination of these cores with particular applications was not taken into account, as seen in the subsequent method. For instance, to combine the AES-ECB core with the HTMS-DAQ system.

To conceal or hide the secret key within the cryptographic core, hardware-software co-designs have been proposed [45], [46], [47], [48], [49]. Nonetheless, a non-pipelined architecture for AES is examined in [45]. This can be achieved with contemporary FPGA chip technology featuring integrated hardware processors. Xilinx provides FPGAs featuring PowerPC hardware cores, while Intel supplies FPGAs equipped with ARM hardware processors. This applies to the Intel Cyclone® V SE 5CSEBA6U23I7 device, featuring 110 thousand Logic Elements (LEs), similar to those present on the Terasic DE10-nano and DE10-Standard boards. The DE10-nano board, for instance, features a serial configuration device, EPCS64 flash memory, USB-Blaster II for onboard programming, 2 push buttons, 4 slide switches, 8 green user LEDs, a 50MHz clock source from the clock generator, and an on-chip ARM-powered Hardware Processor Side (HPS). This is the reason this board has been chosen for the execution and validation of this research work.

Additional researchers examining hardware cores indicated speeds of 2.34 Mb/s with a VLSI chip technology, peaking at 5.3 Gb/s in [50]. It is essential to highlight that certain elevated encryption rates were seen on the AES-128 bits, rather than the AES-256 bits. AES-256 bit key demands additional hardware components and requires more time to function. Certain implementations focused on VLSI or ASIC chips, recognized for their speed compared to FPGAs, though lacking in reconfigurability. A recent research investigation on a pipelined AES-256 core utilizing a Xilinx core indicated an encryption throughput of 1.652 Gb/s [51].

A fully-pipelined AES-128-ECB core was introduced previously [25]. Nonetheless, through ModelSim verifications, the determined maximum encryption rate was calculated to be 69.5 Mb/s. The researchers executed the cycles of the AES-ECB and incorporated pipeline registers to store the output State from each of the 10 rounds, duplicating the hardware for each round 10 times on the Xilinx Spartan-3 FPGA using the Xilinx 13.1 ISE design suite.

ModelSim was utilized to confirm the reported outcomes of a different AES pipelined design intended for 128-bit and 256-bit keys [52]. The specified AES-Core was evaluated for the Altera/Intel Cyclone IV FPGA, which used 7426 LEs. The specified core reported a peak frequency of 96 MHz [52]. The testbench was configured to provide a 256-bit secret key using the known values from the NIST test. The provided secret key was: *603deb1015ca71be-2b73aef0857d7781-1f352c073b6108d7-2d9810a30914dff4*, represented in hexadecimal format. The plaintext is represented in hexadecimal format as *6bc1bee22e409f96-e93d7e117393172a*. These test values are expected to produce an output of 128 bits, represented in hexadecimal format as: *f3eed1bdb5d2a03c-064b5a7e3db181f8*. Even though it appeared to function correctly, it needed 70 clock cycles for a 14-round AES hardware core [52]. This quantity can be decreased by employing the well-established single cycle method, similar to that utilized by the single cycle architecture in MIPS microprocessors.

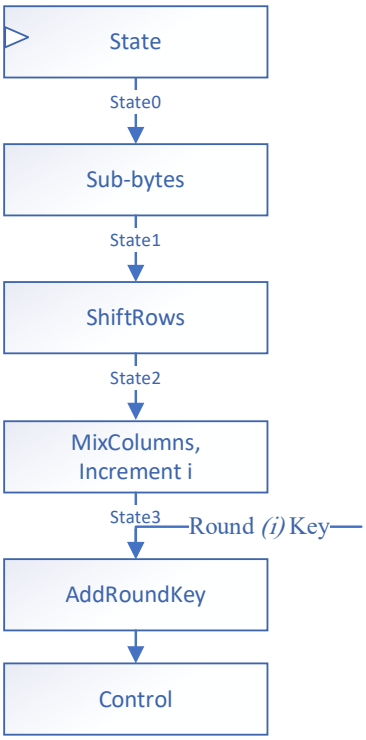

Fig. 5. AES-256-pipelined-Enhanced round architecture using single cycle approach, with different stage registers.

IV. Enhancement of the Fully Pipelined AES-256 Core

In this research, emphasis was on the specified AES-256-Pipelined Core [52], choosing the AES-ECB mode and a key length of 256 bits. The AES-ECB facilitates efficient pipelined execution, and the 256-bit key associated with AES enhances its resistance to Quantum attacks as previously noted. Based on earlier simulation results, the AES-256-pipelined core required 110 cycles to encrypt the initial block. This is attributed to *KeyExpansion* clock cycles and the requirement of 5 clock cycles for every round.

Figure 5 illustrates a single cycle approach that is proposed to decrease this count of clock cycles. Note that the dynamic input indicator (the triangle on the side) signifies a sequential input (i.e., a clock signal), utilized for synchronous modules. In the referenced AES-256-Pipelined core, 5 clock cycles were required of the AES-ECB mode and the key size of 256-bits [52].

The AES-ECB allows for efficient pipelined implementation, and the 256-bit key for AES makes it resilient to Quantum attacks as mentioned before.

In the single cycle approach, several combinational blocks are arranged in sequence, and a single clock regulates the input changes to ensure stability. This method presumes that all the combinational units are linked in series and are capable of finishing their function and achieving a steady-state output within the clock cycle. Typically, extended clock periods are employed for these methods; however, with quicker chip technology, the propagation delay of the combinational blocks may be minimized, allowing the entire operation to be completed within the cycle. This method is implemented in the renowned MIPS single cycle architecture [51] and therefore is utilized in the improvement of the AES-256-Pipelined core.

Figure 5 illustrates the core concept, where one State variable is held in a synchronous register, and each of the combinational blocks produces distinct State variables (for example, State1, State2, etc., as depicted). The control block at the conclusion of every round is tasked with producing the final output (during the last round), or transmitting the intermediate output to the State register.

After careful modification of the given AES-256-Pipelined Core, a successful enhanced core was developed. The next section provides more details on the implantation of both cores.

V. Experimentation, Throughput Comparisons and Cost Comparisons

The DE10-nano board was selected for implementing the AES-256 Core on its FPGA side and the HTMS scheduling on its HPS side. It utilizes a 5CSXFC6D6F31C6 FPGA component, featuring 110k Logic Elements (LE). The DE10-nano board was utilized for the system implementation during the comparisons. Figure 6 illustrates that the FPGA device includes a hardcore processor side (HPS) utilized for executing the software scheduling setup for the HTMS algorithm. The FPGA device's user logic is allocated for the encryption core. This board features a 50MHz crystal oscillator linked to the FPGA pins. The Quartus II software was utilized to perform the simulation and synthesis of all pertinent AES encryption cores for comparative analysis. These are:

- The AES-256-pipelined-core [53]
- Recent research on pipelined AES-256 core [52]
- The Enhanced-AES-256-pipelined-Core (this)

This research's experiment was arranged using a single FPGA device on the DE10-nano board for testing, as shown

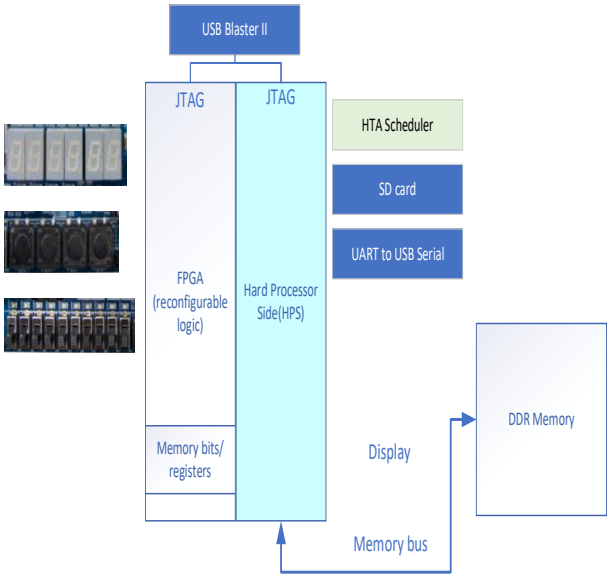

Fig. 6. Proposed single chip system architecture using the FPGA/HPS of DE10-nano board.

in Figure 6. A user within the secure domain operates a computer linked to the FPGA board to configure the secret key of the AES-core via a serial cable for configuration and

TABLE IV:
SUMMARY OF COMPARISON OF RELEVANT FPGA IMPLEMENTATIONS OF AES

Year Available	Model	Tool	Encryption Throughput	FPGA utilization %	Fmax (MHz)	Encryption cycles		
						1st block	Next blocks	Fully Pipelined
this	**Enhanced-AES-256-fully pipelined- single clock cycle -Core (this)**	**ModelSim/ Quartus II, VWF**	**12.8 Gb/s**	**3,664 / 41,910 Logic Blocks (9%)**	**100**	**45**	**14**	**1**
2024	**AES-256-fully pipelined – five cycles [51]**	**Xilinx ISE**	**1.652 Gb/s**	**4864/24576 Slices (19%)**	**180**	**n/a**	**n/a**	**5**
2025	**SymECCipher [53]**	**Xilinx ISE**	**1 Gb/s**	**n/a**	**100**	**n/a**	**n/a**	**n/a**
2021	**AES-256-pipelined-Core [52]**	**ModelSim**	**2.4 Gb/s**	**n/a**	**96**	**110**	**70**	**n/a**
2015	Fully Pipelined AES-128 [41]	Xilinx ISE	69.5 Mb/s	n/a	n/a	n/a	n/a	n/a
2012	Folded Parallel AES-128 [44]	Xilinx Virtex-6 XC6VLX75T	37.1 Gb/s	3788/138240 (2%)	505	2	n/a	n/a

setup. For security purposes, the user's secret key is never transmitted across the network domain.

Table IV presents a summary comparing the relevant AES encryption systems [41], [44], [51], [52], [53]. The emphasis was on improving the AES-256-pipelined-core, not only because it was identified as the quickest but also due to the quantum resistance offered by the AES-256-bit key size. As previously stated, the specified AES-Core utilized 5 clock cycles per round for all 14 rounds of AES-256. Filling the AES pipeline to produce the result of the first input block will require 70 clock cycles (5*14), with an additional 5 clock cycles needed for the output of every subsequent block, assuming the secret key remains unchanged. Additionally, a Figure of Merit (FOM) is created by calculating a weighted sum of the subsequent evaluation criteria.

- Multiplexed, vs dedicated: as the focus of this research is on the time-multiplexed ADC, a factor of 0 is given to dedicated ADC, and a factor of 50 is given to multiplexed ADC systems, to give a relative preference.
- Channel per cost: the given number of channels N is divided by the system's cost and multiplied by a weight of 200. Since this figure combines two features (N, and cost), it is given a relatively higher value.
- Samples per second (Sa/s): Is also given the weight of 200, multiplied by Sa/s (in millions).

TABLE V:
PERFORMANCE COMPARISON OF HTA-BASED DAQ VS. RELEVANT SYSTEMS GIVEN INPUTS IN TABLE II

System	Cost	Scheduling mode	Maximum Sampling rate (Samples/s)	Performance Metrics		
				Utilization U	Efficiency E	Utility Ψ
Prior work (Author with Vaughn) [37])	$58.6⏚	HTA: various computed	1,000,000	100%	100%	100%
Multi-channel DAQ, NI USB-6453, with external analog multiplexer	$4000	Uniform	16,000000	6.25%	100%	100%
ADS 7953EVM-PDK with 16 NXP 74LVC1G53 external analog multiplexers	$257	Uniform	1,000,000	100%	75%	6.25%
Hasan et. al [25]	$44⏚	Various, pre-configured	128,000	n/a	n/a	n/a
Sarker et. al. [31]	$84⏚	Uniform	500	n/a	n/a	n/a
Uktveris et al. [32]	$115⏚	Uniform	1000	n/a	n/a	n/a
Biagetti et. al. [33]	$80⏚	Uniform	9600	n/a	n/a	n/a
Liu et. al [34]	$54	Uniform	500	n/a	n/a	n/a
Choi et. al [30]	n/a	Uniform	100,000,000	n/a	n/a	n/a

⏚ indicates an estimated cost based on literature

Based on these assumptions, the FOM is computed for all systems.

$$FoM = wt_1 * Mux + wt_2 * \frac{N}{\frac{Cost}{100}} + wt_3 * Sa_{per_s}/1{,}000{,}000,$$

where
wt_1=50, wt_2= wt_3=200
$Sa_{per\ s}$ is the Sa/s measure.

The percent increase in FoM is:

$$FoM\%increase = \frac{FoM_i - FoM_j}{FoM_j} * 100, \qquad (9)$$

where FoM_i is for the system of interest, and FoM_j is for the system of comparison. This is represented in Table V. It is worth noting that the author and his research team implemented the HTMS on DSP in [37] for a heterogenous sampling rate of 1MSa/s. The signal quality assessment was done in [37], using inputs of known frequencies given in Table I, and a RMS of 8.9 is reported between the known frequencies and the acquired from the sampled data, for the given 32 channels. This research article extends the usability of the HTMS and integrated with enhanced AES-encryption core of 12.8 Gb/s, which can be used with multi-channel DAQs of 12bit ADC, at 1 G Sa/s.

The calculated FoM% increase is 218%, utilizing (9) in relation to the baseline system presented in [36]. As the Samples/s for the leading two systems is identical in Table V, it's important to highlight that this proposed system is approximately 9 times more affordable. Figure 7 illustrates a visual comparison of the peak operational frequency, FPGA

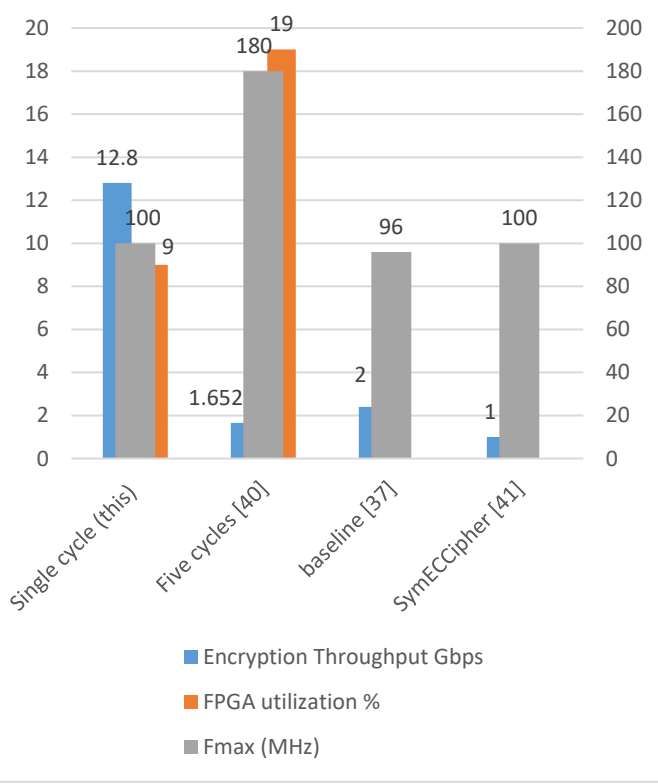

Fig. 7. Graphical comparisons of relevant AES-256 architectures.

device usage, and the maximum encryption throughput attained by the systems being compared, as outlined in Table V. Figure 8 presents the visual comparison of the FoM for the systems that employ multiplexed ADCs listed in Table V.The suggested single-cycle AES core design incorporates stage registers and control signals for the additional combinational blocks within the FPGA, allowing each pipeline stage to need just 1 clock cycle (rather than 5). This consequently raises the utilization of FPGA resources, yet it can be implemented in FPGAs with a high number of LEs, as demonstrated in Table III. Therefore, the initial block requires 28 clock cycles for key expansion, in addition to 14 clock cycles for encryption, and 3 clock cycles prior to the key expansion, amounting to a total of 45 clock cycles. A synthesis-and-fitting process using a computer is carried out on the same FPGA device, and the input oscillator clock frequency of 50 MHz was increased to 100 MHz through embedded Phase Locked Loops (PLL), as the maximum operational frequency was 100 MHz. Table IV presents a summary of the comparisons between the provided core and the proposed enhanced core. When the pipeline operates at full capacity without restarts, the improved Core requires one clock cycle for each subsequent round, completing one full block per cycle. Through equation (5), the throughput is calculated to be:

T_{max}= *128 / (1 *10 ns)*, or T_{max} ***=12.8Gb/s*** for the enhanced Core operating at 100 MHz (i.e. $\tau = 10\,ns$).

This achieved rate of T_{max} exceeds 7 times that of recent research indicating 1.652 Gb/s for AES-256-pipelined architecture on a Xilinx Virtex-4QV XQR4VSX55 FPGA [40]. If the mentioned AES core were to function at its stated maximum clock rate of 96 MHz, its throughput would be T = 128 / (5 * τ), leading to T = 2.4 Gb/s at 96 MHz, which is five times lower than the Tmax for our upgraded AES-256-Core; however, this finding is derived by the author of this paper and has not been shared in their referenced work. The encryption throughput meets the requirements for the suggested HTMS-based DAQ at speeds reaching 12.8 Gb/s, allowing the ADC configured for 12 bits/sample to fulfill KPI3.

VI. CONCLUSIONS AND FUTURE WORK

There are various challenges for multi-channel data acquisition systems, particularly for large number of input channels, which has large deviations in their frequency harmonics. To address them, a hardware reconfigurable board is used to demonstrate the use of FPGA with ADC systems. The Conflicting Time-Samples (CTS) problem is demonstrated in scheduling of heterogenous multi-channel DAQs channels, using conventional scheduling approaches. Then the novel Heterogenous Time-multiplexing Algorithm (HTMS), is proposed and showed how it can be used to dynamically assign a sampling schedule for a single ADC, and avoids conflicting time samples at real-time (KPI2). The steps of the HTMS, are demonstrated along with its feasibility condition, and a numerical example of its working. The HTMS was implemented on a custom hardware using a DSP for 32-channels, and evaluated its performance as it compares with other relevant systems. The proposed algorithm is shown to outperform uniform/homogenous scheduling as it reaches 100% of ADC's utilization and utility, while **other algorithms**

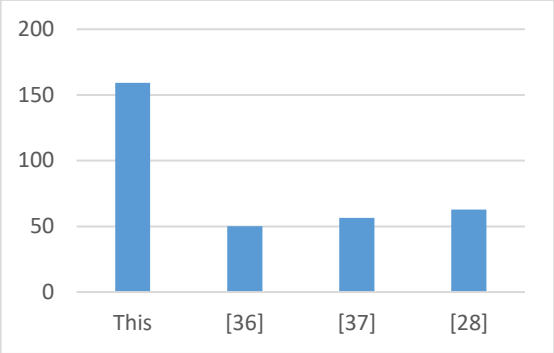

Fig. 8. FoM for relevant multiplexed ADC systems.

for the same set of input sets reduce to 75% utilization, and 6.25% utility. A FoM equation is devised to summarize the features of comparisons across selected systems. Furthermore, the devised FoM of the proposed system is 218% more than the FoM of the nearest system of comparison that uses a multiplexed ADC, while being about 9 times cheaper than the same system (KPI3).

In this paper, various AES cores (folded, parallel, pipelined, hardware-software codesign, etc.) were compared and emphasis was placed on hardware reconfigurable implementations of the Advanced Encryption Standard of 256-bit key due to its known advantages. The provided AES-256-pipelined-core was improved by augmenting the count of combinational blocks linked in series to function at the identical clock cycle for all 14 rounds of the AES-256 algorithm, thereby decreasing the clock cycles from 5 to 1. Results of the comparison (throughput and FPGA utilization) are shown for the improved design in relation to the chosen specified cores. This research reports a maximum throughput of 12.8Gb/s, over 7 times faster (KPI1) than the throughput of comparable recent research findings [51]. The proposed design guarantees that the cryptographic secret key is hidden (i.e., kept internally) within the FPGA device once the user sets up the core at the start of the encryption session.

The key contributions of this research work are:

For a single multi-channel ADC with diverse sampling requirements a novel algorithm for automatic configuration of sampling times is developed and analyzed. Then:

- Using a case study to compare the performance of the proposed algorithm with conventional fixed sampling time techniques, analysis shows a favorable increase in the utilization and utility of the single ADC,
- A reduction in the generated data with the same resolution of all channels,
- A reduction in a DAQ cost and power consumption as compared to state of the art.
- An improved throughput for an enhanced model of a reconfigurable AES-256-bit core using FPGA device by a factor more than 7, by achieving a single cycle per pipeline stage, operating at 100 MHz,
- A developed proposed Enhanced AES-256-Pipelined core in HTMS-based DAQ on a single FPGA chip.
- An extension of the usability of the HTMS and integration with the proposed enhanced AES-encryption core of 12.8 Gb/s, which can be used with multi-channel DAQs of 12bit ADC, at 1 G Sa/s.

FUNDING

This research is funded by Science Technology Development Fund (Egypt) #45945.

REFERENCES

[1] K.K. Sharma, S.D. Joshi, S. Sharma, Advances in Shannon sampling theory. Def. Sci. J. 63(1), 41–45 (2013)

[2] James B. Tsui, “Analog-to-Digital Converters,” Book Chapter, Digital Techniques for Wideband Receivers, p. 155-217, IET Library, 2004.

[3] Yang, Yu-Hang, Jin-Gang Liu, and Shen-Min Song. "A Recursive Non-Uniform Sampling Estimator for Asynchronous Nonlinear Systems" *Sensors*. 2024, 24, no. 9: 2882 (2024). https://doi.org/10.3390/s24092882

[4] Xu, L., Li, W. Nonuniform sampling and reconstruction of Diracs signal associated with linear canonical transform and its application. *EURASIP J. Adv. Signal Process.* 2023, 90 (2023). https://doi.org/10.1186/s13634-023-01051-w

[5] Jinming Ma, Yixiao Yang, Ran Tao, Gang Li, Chang Gao, “Spectrum analysis for nonuniform sampling of bandlimited and multiband signals in the fractional Fourier domain,” *Signal Processing,* v. 222, 2024,109539, ISSN 0165-1684, https://doi.org/10.1016/j.sigpro.2024.109539, 2024.

[6] Acuna Briceno, Rafael Andres, “A Scalable FPGA-Based Multi-Channel Data Acquisition System for Parallel Plate Ionization Chamber,” ProQuest Dissertation, 2021.

[7] Jian Lu, Feng Liu, Yingna Guo, “Design of multi-channel data acquisition and transmission system,” 2022 4th International Conference on Intelligent Control, Measurement and Signal Processing (ICMSP), Hangzhou, China, 2022, pp. 729-732, doi: 10.1109/ICMSP55950.2022.9859158.

[8] Junyoung Lee, Youngmo Ku, Sehoon Choi, Goeun Lee, Taehyeon Eom, Hyun Su Lee, Jae Hyeon Kim, Chan Hyeong Kim, “Development of simultaneous multi-channel data acquisition system for large-area Compton camera (LACC*),” Nuclear Engineering and Technology,* v. 55, no. 10, 2023, pp. 3822-3830, ISSN 1738-5733, https://doi.org/10.1016/j.net.2023.06.048

[9] Xiangning Guo, Xinding Zhang, Jianmin Hu, Zhenyu Deng, “A Multi-Channel Millimeter-Wave Imaging High-Speed Data Acquisition and Transmission System Based on Optical Fiber and PCIe Interface,” 2024 *3rd International Conference on Artificial Intelligence and Computer Information Technology (AICIT),* pp. 1-5, 9/2024.

[10] Na Liu, Ran Tao, Robert Wang, Yunkai Deng, Ning Li, Shuo Zhao, “Signal reconstruction from recurrent samples in fractional Fourier domain and its application in multichannel SAR,” *Signal Processing*, v. 131, 2017, pp. 288-299, ISSN 0165-1684, https://doi.org/10.1016/j.sigpro.2016.08.008.

[11] Rachana Rajpal, Hitesh Mandaliya, Jignesh Patel, Praveena Kumari, Pramila Gautama, Vismaysinh Raulji, Praveenlal Edappalaa, H.D Pujaraa, “Embedded multi-channel data acquisition system on FPGA for Aditya Tokamak,” *Fusion Engineering and Design*, 112 (2016) pp. 964–968 .

[12] Lei Qian, Xinpeng Xing, Haigang Feng and Georges Gielen, “Digital Background Calibration Techniques for 2GS/s 12bit Time-Interleaved Pipelined ADC,” Journal of Physics: Conference Series 2613 012020, pp. 1-14, 2023, DOI 10.1088/1742-6596/2613/1/012020.

[13] Niu Guangshan, Liu Cong, Zhang Jianwei, Li Xuetao, Luo Xiangdong,“Research progress of time-interleaved analog-to-digital converters,” Integration, Volume 81, Pages 313-321, ISSN 0167-9260, https://doi.org/10.1016/j.vlsi.2021.08.007, 2021.

[14] D'Arco, et. Al., “Enabling Fine Sample Rate Settings in DSOs with Time-Interleaved ADCs,” Sensors, Volume22, Issue1, DOI10.3390/s22010234, 2021.

[15] Chen, Dongdong Cui, Xinhui Zhang, Qidong Li, Di Cheng, Wenyang Fei, Chunlong Yang, Yintang, “A Survey on Analog-to-Digital Converter Integrated Circuits for Miniaturized High Resolution Ultrasonic Imaging System,” Journal of Micromachines, Volume13, Issue1, DOI 10.3390/mi13010114, 2022.

[16] Adimulam, Mahesh Kumar, Srinivas, M. B., “A 12-bit, 1.1-GS/s, Low-Power Flash ADC,” IEEE Transactions on Very Large Scale Integration (VLSI) Systems, DOI10.1109/TVLSI.2021.3138538, 2022.

[17] Luengo-Garcia, David, Pantaleon-Peieto, Carlos, Santamaria Caballero, Jgnacio, Jesus, Ibanex Diaz, and Gomez-Cosio, Enrique. "Simultaneous Sampling by Digital Phase Correction," IEEE Instrumentation and Measurement, 1997.

[18] R. R. M. Meurer, "Enhancement of Multichannel A/D Conversion by a Code Division Multiplex Approach," IEEE 6th International Symposium, 2000.

[19] Moon, Jim Visser, Henk Hunt, Robert Kenneth McCombie, Devin Dhillon, Marshal Singh Banet, Matthew J., "Body-worn vital sign monitor," USA Patent US-11096596, 2021.

[20] Ryu, Jae-young, Kim, Dong-ki, Kim, Young-hwan, Seo, Min-woo, Lee, Jei-young, "Ultrasound diagnosis apparatus for self-diagnosis and remote-diagnosis, and method of operating the ultrasound diagnosis apparatus," USA Patent US- 11123041, 2021.

[21] Banet, Matthew, Pede, Susan Meeks Dhillon, Marshal Singh Hunt, Kenneth Robert, "Body-worn sensor for characterizing patients with heart failure," USA Patent US- 11129537, 2021.

[22] Islam, Mohammed N., "Multi-wavelength wearable device for non-invasive blood measurements in tissue," Patent US- 11160455, 2021.

[23] Saeed, Maryam, et. al, "Evaluation of Level-Crossing ADCs for Event-Driven ECG Classification," IEEE transactions on biomedical circuits and systems, DOI 10.1109/TBCAS.2021.3136206, 2021.

[24] Somappa, Laxmeesha, Baghini, Maryam Shojaei, "Continuous-Time Hybrid Delta Sigma Modulators for Sub-mu W Power Multichannel Biomedical Applications," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, doi 10.1109/TVLSI.2022.3140222, 2022.

[25] O. Alkelany, "Simultaneous Time Sampling for Heterogenous Multichannel Data Acquisition," International Journal of Advanced Scientific Research and Innovation. Vol. 4, no. 2, 2021, pp 39 – 52. ISSN: 2785-9541, 2021

[26] O. Alkelany M. Abdallah, A. Alouani; "A Low-Cost Stand-Alone Multichannel Data Acquisition, Monitoring, and Archival System with On-Chip Signal Processing," IEEE Transaction on Instrumentation and Measurement, vol. 60 (8), pp. 2813-2827, DOI: 10.1109/tim.2009.2036402, 2011.

[27] Samit Hasan, Tanvir Pantha, Muhammad Abdullah Arafat, "Design and development of a cost-effective portable IoT enabled multi-channel physiological signal monitoring system," Biomedical Engineering Advances, vol. 7, pp. 1-14, ISSN 2667-0992, https://doi.org/10.1016/j.bea.2024.100124, 2024.

[28] J. Wu, J.-C. Liu, and W. Zhao, "Utilization-Bound Based Schedulablity Analysis of Weighted Round Robin Schedulers," Real-Time Systems Symposium, pp. 435,436, 2007 Dec 2007.

[29] J. W. L. C. L. Liu, "Scheduling Algorithms for Multiprogramming in a Hard-Real- Time Environment," Journal of the ACM, vol. 20, no. 1, pp. 46-61, 1973.

[30] Choi, Soyeon, Heehun Yang, Yunjin Noh, Giyoung Kim, Eunsang Kwon, and Hoyoung Yoo. 2024. "FPGA-Based Multi-Channel Real-Time Data Acquisition System" *Electronics,* vol. 13, no. 15: 2950. https://doi.org/10.3390/electronics13152950

[31] V.K. Sarker, M. Jiang, T.N. Gia, A. Anzanpour, A.M. Rahmani, P. Liljeberg, "Portable multipurpose bio-signal acquisition and wireless streaming device for wearables," 2017 IEEE Sensors Applications Symposium (SAS), pp. 1–6, 2017.

[32] T. Uktveris, V. Jusas, "Development of a modular board for EEG signal acquisition," Sensors, vol. 18 (7), 2018.

[33] G. Biagetti, P. Crippa, L. Falaschetti, C. Turchetti, "A multi-channel electromyography, electrocardiography and inertial wireless sensor module using Bluetooth low-energy," Electronics, vol. 9, no. 6: 934, 2020.

[34] S.-H. Liu, J.-J. Wang, T.-H. Tan, "A portable and wireless multi-channel acquisition system for physiological signal measurements," Sensors, vol. 19, no. 23, 2019.

[35] Abdallah, M. A. S. (2010). Reconfigurable stand-alone multi-channel data acquisition, archiving, monitoring and processing system using FPGA-SOPC technology with heterogeneous maximal service scheduling (Order No. 3418848). ProQuest Dissertations & Theses Global. (751247740).

[36] O. Alkelany, M. Abdallah, "Heterogeneous Maximal Service Scheduling Technique," US61/326,725 USA Prov. Patent, 2011.

[37] Vaughn, M. B. (2013). A low cost standalone multichannel data acquisition system utilizing heterogeneous time-multiplexed sampling (Order No. 1553996). ProQuest Dissertations & Theses Global. (1525998314).

[38] (NIST), N. , Dworkin, M. , Sonmez Turan, M. and Mouha, N. (2023), Advanced Encryption Standard (AES), Federal Inf. Process. Stds. (NIST FIPS), National Institute of Standards and Technology, Gaithersburg, MD, [online], https://doi.org/10.6028/NIST.FIPS.197-upd1, https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=936594 accessed March, 2026.

[39] Adam J. Elbirt, W. Yip, B. Chetwynd, and C. Paar, "An FPGA-Based Performance Evaluation of the AES Block Cipher Candidate Algorithm Finalists," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 9(4), pp. 545 – 557, doi: 10.1109/92.931230, 2001.

[40] Chitu, Cristian, Glesner, Manfred, "An FPGA implementation of the AES-Rijndael in OCB/ECB modes of operation", *Microelectronics Journal*, vol. 36(2), pp. 139-146, 2005.

[41] A. P. Anusha Naidu and P. K. Joshi, "FPGA implementation of fully pipelined Advanced Encryption Standard," 2015 International Conference on Communications and Signal Processing (ICCSP), Melmaruvathur, India, 2015, pp. 0649-0653, doi: 10.1109/ICCSP.2015.7322568.

[42] Rouvroy, Gael, Standaert, Francois-Xavier, Quisquater, Jean- Jacques, Legat and Jean-Didier, "Compact and efficient encryption/decryption module for FPGA implementation of the AES Rijndael very well suited for small embedded applications", *International Conference on Information Technology*: Coding Computing, ITCC'04, vol. 2, p 583-587, 2004.

[43] G. Umamaheswari and A. Shanmugam, "Efficient VLSI implementation of the block cipher Rijndael algorithm", *Academic Open Internet Journal, http://www.acadjournal.com/* vol. 12, 2004.

[44] K. Rahimunnisa, P. Karthigaikumar, Soumiya Rasheed, J. Jayakumar, S. SureshKumar, "FPGA implementation of AES algorithm for high throughput using folded parallel architecture," 2012, Wiley, Security and Communication Networks, doi: https://doi.org/10.1002/sec.651

[45] Dong, L., Ye, X., Zhuang, L., Zhan, R., & Hossain, M. S. (2024). AES software and hardware system co-design for resisting side channel attacks. Expert Systems, 41(10), e13664.

[46] Lata, K., Chhabra, S., & Saini, S. (2021). Hardware–software Co-design framework for data encryption in image processing Systems for the Internet of things environment. IEEE Consumer Electronics Magazine, 11(4), 92–97.

[47] Jati, A., Gupta, N., Chattopadhyay, A., & Sanadhya, S. K. (2024). A configurable crystals-kyber hardware implementation with side-channel protection. ACM Transactions on Embedded Computing Systems, 23(2), 1–25.

[48] Dubey, A., & Aysu, A. (2023). A full-stack approach for Side-Channel secure ML hardware. In 2023 IEEE international test conference (ITC) (pp. 186–195). IEEE.

[49] Buhan, I., Batina, L., Yarom, Y., & Schaumont, P. (2022). Sok: Design tools for side-channel-aware implementations. In Proceedings of the 2022 ACM on Asia conference on computer and communications security. (pp. 756–770).

[50] R. Kumar, S. Taneja, V. De and S. K. Mathew, "A 4.7-to-5.3-Gb/s Fault-Injection and Side-Channel Attack-Resistant AES-256 Engine Using Masked Isomorphic Composite Fields in Intel 4 CMOS," in IEEE Journal of Solid-State Circuits, vol. 60, no. 4, pp. 1349-1358, April 2025, doi: 10.1109/JSSC.2025.3541573.

[51] Assaad EL Makhloufi, Samir EL Adib, Naoufal Raissouni, " Hardware pipelined architecture with reconfigurable key based on the AES algorithm and hamming code for the earth observation satellite application: Sentinel-2 satellite data case," *e-Prime - Advances in Electrical Engineering, Electronics and Energy,* Volume 8, 2024, 100548, ISSN 2772-6711, https://doi.org/10.1016/j.prime.2024.100548. https://www.sciencedirect.com/science/article/pii/S277267112400130X

[51] A. E. Oghostinos, K. Moussa, A. Elnaggar, A. AbdAlRhman and A. Soltan, "Single-Cycle MIPS Processor based on Configurable Approximate Adder," 2022 11th International Conference on Modern Circuits and Systems Technologies (MOCAST), Bremen, Germany, 2022, pp. 1-4, doi: 10.1109/MOCAST54814.2022.9837642

[52] MichaleHab, "AES-Core", https://github.com/michaelehab/AES-Verilog, accessed March, 2021. 256-bit AES-ECB

[53] Selvi, P., Sakthivel, S. A hybrid ECC-AES encryption framework for secure and efficient cloud-based data protection. Sci Rep 15, 30867 (2025). https://doi.org/10.1038/s41598-025-01315-5

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- rms.xlsx