ELECTRONICS AND ELECTRICAL ENGINEERING
ISSN 1392 – 1215 2007. No. 6(78)
ELEKTRONIKA IR ELEKTROTECHNIKA

SIGNAL TECHNOLOGY
T121
SIGNALŲ TECHNOLOGIJA

Development of Asynchronous Data Processing System by using General-purpose Microprocessors

A. Baums, M. Greitans, U. Grunde
Institute of Electronics and Computer Science,
Dzerbenes st. 14, LV-1006, Riga, Latvia, phone: +371 67558134; e-mail: baum@edi.lv

Introduction

The asynchronous data processing is a method that was already used in the computers of the first generation. For example asynchronous data processing was successful used in the first in Latvia built computer LM-3 [1]. The revival of asynchronous processing started in nineties of the past century. The key benefits of asynchronous system [2] are very low power consumption, absence of the clock screw, reduced heat elimination, low electromagnetic emission, adaptation to physical properties.

Increasing number of applications using distributed or autonomous data processing systems are looking for micro-systems with micro-energy consumption and high performance at the same time. Several existing techniques can be used to reach these targets including asynchronous design [2], power consumption control [3] and non-conventional digital signal processing methods [4, 5].

Asynchronous design approach is widely used at the functional level to design interface between units with different cycle time. In recent years asynchronous design approach has been used to create clock-less architectures [6]. Several successful developments has been announced [7,8] but till now no off-shelf asynchronous microprocessor or A/D converter is available. In practice asynchronous approach still remains on the functional level.

Power control has becoming efficient tool to decrease energy consumption. It includes power scheduling and dynamic voltage scaling. Power scheduling minimizes energy consumption by turning-off resources not in use. Dynamic voltage scaling is controlling supply voltage and performance of a microprocessor. The most energy savings may be achieved in power-down mode. Advanced microprocessors have several power reduction modes including power-down mode. Power reduction may be also achieved by decreasing clock frequency than can be done by software.

The non-conventional digital signal processing methods which are targeting non-stationary signals often are implementing signal-driven sampling instead clock-driven sampling that is provides equally spaced samples. Signal-driven sampling means that signals are sampled only when measured signal pass certain predetermined reference level. Usually these limits are predetermined by reference levels. It is named also level-crossing sampling [9]. That can give decreased number of data compare with uniform sampling and allows to use system resources only when it is necessary.

Implementation of principles of power consumption control and level-crossing sampling into general-purpose microprocessors can be used to develop data processing system that has features similar to asynchronous systems. Such a standstill system will consume almost nothing. It will stay in such condition till signal at the input has reached first predetermined reference level so-called start-up level.

Micro-power general-purpose microprocessor and data level-crossing sampling is proposed to overcome the low power asynchronous data processing system design problems. The problems and the first results of such system development are proposed in this article.

Principles and architecture of asynchronous data processing system

Implementation of the principles of the data level-crossing sampling, switching system to power reduction modes and asynchronous interaction between system units is determining asynchronous nature of the data processing system. These are basic principles to ensure very low energy consumption and provide necessary performance at the same time.

Two following measurements of level-crossing sampling are never equal if the signals slope keeps the sign

$$x(t_i) \neq x(t_{i+1}), \quad (1)$$

where $n = 1 \div N$ (N is number of predetermined reference values). The relationship (1) is not true when signal slope changes the sign. This feature is used to detect the change of the signal slope sign. Usually a difference between two following reference values is fixed. In our case it may be not fixed. It is the case when new reference value is taken from predetermined set of reference values instead of

calculating it from previous reference value. Such approach increases flexibility and decreases calculation time of the reference value. Usually two reference level values r_n^+ and r_n^- are used to compare with the input signal. The system is tracking the input signal according to (2).

$$r_n^- < x(t_i) < r_n^+ . \quad (2)$$

At the time instants when $x(t_n) = r_n^-$ or $x(t_n) = r_n^+$ the input signal measurement is performed. After the measurement relationship (2) is reestablished. It means that data sampling is tracking changes of the input signal. Any substantial increase or decrease of the input signal will cause increase or decrease of the reference levels. Level-crossing sampling is used to trigger the switching of the asynchronous data processing system from the power-down mode to the active mode. The switching happens if input signal value has reached the start-up reference level value. Crossing of start-up level causes waking-up and energizing of the asynchronous data processing system. Proposed approach to level-crossing sampling is similar to the approach described in [8]. The difference consists in the way how to calculate reference level values. A classical approach [8, 9] suggests that for any n

$$\left(r_n^+\right) - \left(r_n^-\right) = q , \quad (3)$$

where q is quantization step. It means that for n+1 reference level the value will be calculated as

$$\left(r_{n+1}^{\pm}\right) = \left(r_n^{\pm}\right) \pm q , \quad (4)$$

where the choice of a sign is depending from the slope direction. It is a typical way of calculating reference level values. Rewriting (4) as

$$\left(r_{n+1}^{\pm}\right) = f(n \pm 1), \quad (5)$$

where $f(n \pm 1)$ is any function defined for $n = 1 \div N$. Satisfying condition

$$q \le f(n \pm k) \le Nq \quad (6)$$

Fig. 1 shows the architecture of the system based on the principles of the data level-crossing sampling. Basic elements include two comparators, two Digital-to-Analog Converters (DAC) and a timer. Input signal is connected to the input of each comparator. DAC1 is providing a reference level r^+ for a rising slope of the signal. DAC2 is providing a reference level r^- for a falling slope of the signal. Outputs of DAC1 and DAC2 are connected with respective inputs of the comparator1 and the comparator2. Outputs of both comparators are controlling time measurements by gating the input of the timer. General-purpose microprocessor (μP) controls all basic elements and coordinates data sampling. Major functions of μP are: calculation of reference levels, loading new reference level values in DAC1 and DAC2, processing status of the comparator outputs and saving timer values.

The algorithm of the level-crossing data sampling on Fig. 2 comprises 4 major stages. The initialization stage prepares the system for the start. During this stage μP configures comparators, DAC and timers. The power-down mode is providing power conservation when the system is idle. The asynchronous data processing system switches to the power-down mode if the input signal has not reached start-up level value during time interval τ. During power-down only the comparators and DAC have a power. Comparators are sensing an input signal. DAC are providing the reference level. Data sampling starts only after switching the system from the power-down mode to the active mode. It happens when the input signal reaches the start-up level value. The comparator1 output causes an interrupt that is switching the system to the active mode. After data sampling and data processing the system goes back to the energy saving mode if the input signal is below start-up level value more than time interval τ. This algorithm minimizes power consumption by switching-off the system when the input signal is not active.

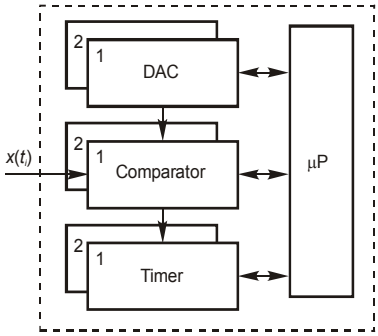

Fig. 1. Architecture of the asynchronous data processing system

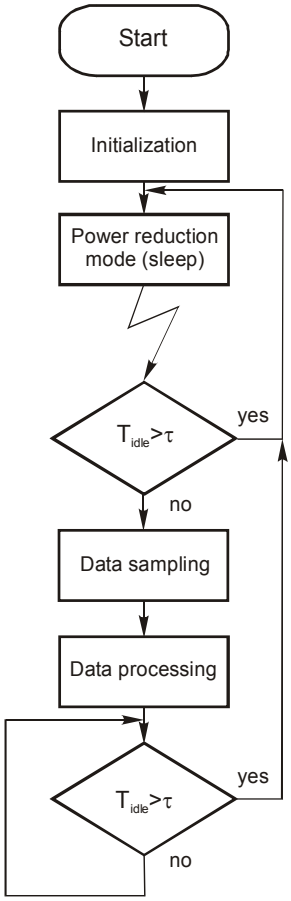

Fig. 2. Algorithm of the data sampling

Implementation

Philips P89PLC936 micro-controller [10] was used to build the prototype of the asynchronous data processing system. The micro-controller has two DAC, two comparators and two timers on a chip that allows to build the prototype system without any additional elements. Philips P89PLC936 micro-controller has advanced power control with 3 energy saving modes. The control software for data sampling and power control was developed in C using the algorithm from Fig.2. Sampled data is saved as sequence of couples (t_n, α_n) where t_n is current timer value and α_n is boolean variable indicating the direction of the slope. The power control of Philips P89PLC936 allows to switch the micro-controller to power-down mode leaving the comparators and DAC powered. Reduction of power consumption may be achieved by decreasing sampling frequency. Time delay for switching from power-down mode to active mode varies from 35 to 94 μs depending from CPU clock frequency.

Evaluation of the prototype system was performed analyzing the relationship between number of sampling levels *N*, maximum signal frequency $f_{\max}$ and maximum loop delay $\delta_{\max}$. The maximum loop delay $\delta_{\max}$ determines the time interval available to prepare for next sampling cycle. It is calculated from the (7) using modified equation from [8]

$$\delta_{\max} = 1/(2\pi f_{\max} N). \quad (7)$$

Calculated different $\delta_{\max}$ values for different number of sampling levels *N* and maximum frequencies of different input signal $f_{\max}$ are given in Table 1. These values are setting limits in terms of *N* and $f_{\max}$ for the data processing system.

Table 1. Maximum loop delay values for different *N* and $f_{\max}$

$f_{\max}$ / N	2kHz input signal loop delay μs	1kHz input signal loop delay μs	0.5kHz input signal loop delay μs
31	2.57	5.14	10.27
15	5.31	10.62	21.23
7	11.37	22.75	44.50

There are two CPU clock cycles per μP cycle for P89PLC936 micro-controller. The most instructions are executed in one or two μP cycles. The timer clock cycle is the same as μP cycle. Therefore a difference between two timer measurements is equal to the number of executed μP cycles. It makes sense to express maximum loop delay values in a number of μP cycles. Using (7) and values for *N* = 15 (from Table 1.) K_c is calculated as

$$K_c = \delta_{\max} f_{\mu P} / 2, \quad (8)$$

where K_c is a number of μP cycles fitting in $\delta_{\max}$ at different CPU frequencies.

Microprocessor based system can be used for level-crossing sampling if the input signal maximum loop delay satisfies (9)

$$K_p < K_c, \quad (9)$$

where K_p is number of μP cycles of the program to process level-crossing sampling. Current version of the control program is spending 60 μP cycles for the processing. Calculated values K_c (in Table 2) that are in bold satisfy the condition (9). Optimization of the program code may decrease $\delta_{\max}$ by 20 – 30%. It means that the system with optimized code will be able to process 4 kHz input signal.

Table 2. Maximum loop delay expressed in number of μP cycles

CPU clock MHz	No of μP cycles for 2kHz signal	No of μP cycles for 1kHz signal	No of μP cycles for 0.5KHz signal
7.373	19	39	**78**
12.000	31	**63**	**127**
15.000	39	**79**	**159**
18.000	47	**95**	**191**

To measure maximum loop delay of the system a single step with amplitude equal to Nq and with rising edge equal 0.5 μs was applied to the signal input. The chosen values of level-crossing sampling were: N = 7; q = 20. Accordingly the timer values were sampled and calculated. Fig.3 displays the measurement of $\delta_{\max}$ for level-crossing sampling. The difference between two following time events is equal $\delta_{\max}$. Difference between consecutive timer measurements equals 58 μP cycles. It is 2 μP cycles less than calculated value. It is because the input signal in form of the single step is always ahead of a preset reference level. Therefore output of the comparator1 is always active that causes a single 2-cycle instruction less.

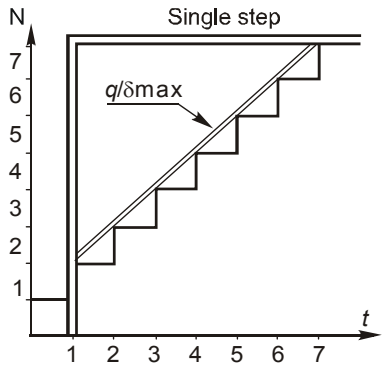

Fig. 3. Measurement using data level-crossing sampling

Usage of a general-purpose μP in the system brings more flexibility and new opportunities in a research of asynchronous data processing system. Availability and ongoing improvement of μP makes them as attractive and easily used tool. At the same time use of μP is limiting the input signal bandwidth of the data level-crossing sampling.

Conclusions

1. The evaluation of the prototype of the asynchronous data processing system built on Philips P89PLC936 micro-controller using a principle of the level-crossing sampling confirms feasibility of such a system.
2. The current version of he asynchronous data processing system is able to process input signal with maximum frequency f_{max} = 2kHz using sampling level number N = 7.
3. Limiting factors of the system are maximum loop delay and time delay for switching from power-down mode to active mode.

References

1. **Даубе Я., Баумс А.** Электронная дискретная вычислительная машина LM–3 // Автоматика и вычислительная техника. – 1961. – № 1. – С. 99–112.
2. **Sparsø J., Furber S.** Principles of asynchronous circuit design – A systems perspective, Kluwer Academic Pb. – 2001.
3. **Baums A.** Energy Consumption Optimization in Hard Real–Time System CMOS Processors // Electronics and Electrical Engineering. – 2006. – No. 4(68). – P. 19–22.
4. **Greitans M., Homjakovs I.** Enhanced Digital Signal Processing of Signal–Dependently Sampled Signals // Electronics and Electrical Engineering. – 2006. – No. 4(68). – P. 9–14.
5. **Greitans M.** Processing of Non–Stationary Signal Using Level–Crossing Sampling // Proceedings of the International Conference on Signal Processing and Multimedia Applications "SIGMAP2006" Setubal, Portugal, Aug. 2006. – P. 170–177.
6. **Ekanayake V., Kelly IV C., Manohar R.** SNAP: A Sensor Network Asynchronous Processor // Proceedings of International Symposium on Asynchronous Circuits and Systems ASYNC'03. – 2003.
7. **Brunvand E., Nowick S., Yun K. Practical advances in asynchronous design and in asynchronous/synchronous interfaces** // Proceedings of the 36th ACM/IEEE conference on Design Automation. – 1999. – P. 104–109.
8. **Allier E., Sicard G., Fesquet L., Renaudin M.** A new class of asynchronous a/d converters based on time quantization // Proceedings of International Symposium on Asynchronous Circuits and Systems ASYNC'03. – 2003. – P. 196–205.
9. **Allier E., Goulier J., Sicard G., Dezzani A., André E., Renaudin M.** A 120 nm Low Power Asynchronous ADC // International Symposium on Low Power Electronics and Design ISLPED'05. – 2005. – P. 60–65.
10. Koninklijke Philips Electronics N.V. P89LPC/933/934/935/ 936 User Manual Rev.2 – 2005.

Submitted for publication 2007 03 05

A. Baums, M. Greitans, U. Grunde. Development of Asynchronous Data Processing System by using General-purpose Microprocessors // Electronics and Electrical Engineering. – Kaunas: Technologija, 2007. – No. 6(78). – P. 21–24.

Development of asynchronous data processing system is usually based on dedicated asynchronous elements. Advanced general-purpose microprocessors have a set of features facilitating development of asynchronous data processing system. Such a system is designed for non-stationary signals using non-conventional digital signal processing methods. Implementation of the principles of the data level-crossing sampling, switching system to power reduction modes and asynchronous interaction between system units is determining asynchronous nature of the system. Paper represents architecture of the system implemented on Philips P89PLC936 micro-controller using a principle of the level-crossing sampling. The system is able to process input signal with maximum frequency f_{max} = 2kHz using sampling level number N = 7. Ill. 3, bibl. 10 (in English, summaries in English, Russian and Lithuanian).

А. Баумс, М. Грейтанс, У. Грунде. Разработка асинхронной системы обработки данных на основе микропроцессоров общего применения // Электроника и электротехника. – Каунас: Технология, 2007. – № 6(78). – С. 21–24.

Разработка асинхронной системы обработки данных основывается обычно на специализированных асинхронных элементах. Развитые микропроцессоры общего применения имеют набор свойств, облегчающих разработку асинхронной системы обработки данных. Система такого рода предназначена для нестационарных сигналов, используя численные методы обработки сигналов. Осуществление принципов сбора данных пересечением уровня, переключение системы в режим низкого потребления и асинхронное взаимодействие между элементами системы определяет асинхронные свойстсва системы. В статье описывается архитектура системы реализована на Philips микроконтроллере P89PLC936 используя принцип сбора данных пересечением уровня. Система в состоянии обработать входной сигнал с максимальной частотой 2 кГц при числе уровней - 7. Ил. 3, библ. 10 (на английском языке; рефераты на английском, русском и литовском яз.).

A. Baums, M. Greitans, U. Grunde. Asinchroninių duomenų apdorojimo sistemų kūrimas naudojant mikroprocesorius // Elektronika ir elektrotechnika. – Kaunas: Technologija, 2007. – Nr. 6(78). – P. 21–24.

Asinchroninės duomenų perdavimo sistemos dažnai yra kuriamos specializuotų asinchroninių elementų pagrindu. Bendros paskirties mikroprocesoriai turi savybių, palengvinančių kurti asinchronines duomenų apdorojimo sistemas. Paprastai tokia sistema, skirta nestacionarių signalų skaitmeniniam apdorojimui, remiasi nestandartiniais metodais. Duomenų lygių perėjimo atrankos algoritmas naudingas dėl mažų energijos sąnaudų. Straipsnyje aprašyta sistema su Philips P89PLC936 mikrokontroleriu, sudaryta naudojant duomenų lygių perėjimo atrankos algoritmą. Sistema sugeba apdoroti maksimalaus f_{max} = 2kHz dažnio signalus, kai lygių skaičius N = 7. Il. 3, bibl. 10 (anglų kalba; santraukos anglų, rusų ir lietuvių k.).

FURTHER READING

Click any one of the following links to be taken to a website which contains the following documents.

The following are some recent examples of Asynchronous ADC activity off the web.

6 bit Asynchronous December 2006
Asynchronous ADC In CAD Mentor Graphics
Asynchronous Data Processing System
ASYNCHRONOUS PARALLEL RESISTORLESS ADC
Flash Asynchronous Analog-to-Digital Converter
Novel Asynchronous ADC Architecture
LEVEL BASED SAMPLING FOR ENERGY CONSERVATION IN LARGE NETWORKS
A Level-Crossing Flash Asynchronous Analog-to-Digital Converter
Weight functions for signal reconstruction based on level crossings
Adaptive Rate Filtering Technique Based on the Level Crossing Sampling
Adaptive Level—Crossing Sampling Based DSP Systems
A 0.8 V Asynchronous ADC for Energy Constrained Sensing Applications
Spline-based signal reconstruction algorithm from multiple level crossing samples
A New Class of Asynchronous Analog-to-Digital Converters
Effects of time quantization and noise in level crossing sampling stabilization

Here is some more background information on Analog to Digital converters.

A 1-GS/s 6-bit 6.7-mW ADC
A Study of Folding and Interpolating ADC
Folding_ADCs_Tutorials
high speed ADC design
Investigation of a Parallel Resistorless ADC

Here are some patents on the subject.

4,291,299_Analog_to_digital_converter_using_timed
4,352,999_Zero_crossing_comparators_with_threshold
4,544,914_Asynchronously_controllable_successive_approximation
4,558,348_Digital_video_signal_processing_system_using
5,001,364_Threshold_crossing_detector
5,315,284_Asynchronous_digital_threshold_detector_
5,945,934_Tracking_analog_to_digital_converter
6,020,840_Method_and_apparatus_for_representing_waveform
6,492,929_Analogue_to_digital_converter_and_method
6,501,412_Analog_to_digital_converter_including_a_quantizers
6,667,707_Analog_to_digital_converter_with_asynchronous_ability
6,720,901_Interpolation_circuit_having_a_conversio2
6,850,180_SelfTimed_ADC
6,965,338_Cascade_A_D_converter
7,133,791_Two_mean_level_crossing_time_interval

11.19.10_1.20PM
dsauersanjose@aol.com
Don Sauer