POLSKA AKADEMIA NAUK
KOMITET ELEKTRONIKI I TELEKOMUNIKACJI

KWARTALNIK ELEKTRONIKI I TELEKOMUNIKACJI

ELECTRONICS AND TELECOMMUNICATIONS QUARTERLY

TOM 52 — ZESZYT 1

WARSZAWA 2006

ADRES REDAKCJI
00-665 Warszawa, ul. Nowowiejska 15/19 Politechnika, pok. 470
Instytut Telekomunikacji, Gmach im. prof. JANUSZA GROSZKOWSKIEGO

Dyżury Redakcji: środa i piątki, godz. 14–16
tel. (022) 660 77 37

Telefony domowe: Redaktora Naczelnego: 812 17 65
Zast. Red. Naczelnego: 826 83 41
Sekretarza Odpowiedzialnego: 633 92 52

Ark. wyd. 10,0 Ark. druk. 8,5	Podpisano do druku w styczniu 2006 r.
Papier offset. kl. III 80 g. B-1	Druk ukończono w styczniu 2006 r.

Skład, druk i oprawa: Warszawska Drukarnia Naukowa PAN
00-656 Warszawa, ul. Śniadeckich 8
Tel./fax: 628-87-77

Szanowni Autorzy

„Kwartalnik Elektroniki i Telekomunikacji" — Electronics and Telecommunications Quartery jest kontynuatorem tradycji powstałego 52 lata temu kwartalnika p.t. „Rozprawy Elektrotechniczne".

Kwartalnik jest czasopismem Komitetu Elektroniki i Telekomunikacji Polskij Akademii Nauk. Wydawany jest przez Warszawską Drukarnię Naukową PAN. Kwartalnik jest czasopismem naukowym, na którego łamach są publikowane artykuły i komunikaty prezentujące wyniki oryginalnych prac teoretycznych i doświadczalnych, a także przeglądowych. Związane są one z szeroko rozumianymi dziedzinami współczesnej elektroniki, telekomunikacji, mikroelektroniki, oproelektroniki, radiotechniki i elektroniki medycznej.

Autorami publikacji są wybitni naukowcy, znani specjaliści o wieloletnim doświadczeniu, a także młodzi badacze — głównie doktoranci.

Artykuły charakteryzują się oryginalnym ujęciem zagadnienia, interesującymi wynikami badań, krytyczną oceną teorii lub metod, omówieniem aktualnego stanu, lub postępu danej gałęzi techniki oraz omówieniem perspektyw rozwojowych. Sposób pisania matematycznej części artykułów zgodny jest z wytycznymi IEC (International Electronics Commision) oraz ISO (International Organization of Standarization).

Wszystkie publikowane w Kwartalniku artykuły są recenzowane przez znanych krajowych specjalistów, co zapewnia że publikacje te są uznawane jako autorski dorobek naukowy. Opublikowane w kwartalniku wyników prac naukowych zrealizowanych w ramach „GRANTów" Komitetu Badań Naukowych spełnia więc jeden z wymogów stawianych tym pracom.

Czasopismo dociera do wszystkich zajmujących się elektroniką i telekomunikacją krajowych ośrodków naukowych oraz technicznych, a także szeregu instytucji zagranicznych. Jest ponadto prenumerowane przez liczne grono specjalistów i biblioteki.

Każdy Autor otrzymuje bezpłatnie 20 egzemplarzy nadbitek swojego artykułu, co ułatwia przesłanie go do indywidualnych wybranych przez Autora osób i instytucji w kraju lub za granicą. Ułatwia to dodatkowo fakt, że w Kwartalniku są publikowane artykuły w języku angielskim.

Nadesłane do redakcji artykuły są publikowane w terminie około pół roku, w przypadku sprawnej współpracy Autora z Redakcją. Wytyczne dla Autorów dotyczące formy publikacji są zamieszczone w zeszytach Kwartalnika, można je także otrzymać w siedzibie Redakcji.

Artykuły można dostarczać osobiście, lub pocztą pod adresem zamieszczanym na stronie redakcyjnej w każdym zeszycie.

Redakcja

SPIS TREŚCI

CONTENTS

KWARTALNIK ELEKTRONIKI I TELEKOMUNIKACJI, 2006, 52, z. 1, ss. 7–22

Influence of the laser turn-on delay jitter on BER performance in telecommunication links

ŁUKASZ ŚLIWCZYŃSKI, PRZEMYSŁAW KREHLIK

Departament of Electronics, AGH University of Scence and Technology
Mickiewicza 30 Ave., 30-059 Kraków
sliwczyn@uci.agh.edu.pl,
krehlik@uci.agh.edu.pl

Otrzymano 2004.10.01
Autoryzowano 2004.12.20

Paper presents the analysis of sensitivity penalty induced by turn-on delay jitter resulting from subthreshold biasing of directly modulated semiconductor laser. Modelling of the laser turn-on delay used in the analysis assumes that Auger process dominates the recombination of carriers in the subthreshold region. Bit error rate calculation were performed for different modulation speeds and biasing conditions. The change of the received signal mean value was also taken into account. Obtained results show that subthreshold biasing of semiconductor lasers is possible with reasonable sensitivity penalty even for data rates as high as 2.5 Gb/s and that it is generally not a problem for data rates below 622 Mb/s.

Keywords: sensitivity penalty, laser subthreshold biasing, laser turn-on delay, TDMA, bit error rate

1. INTRODUCTION

Subthreshold laser biasing is attractive in some transmission systems because it allows achieving high extinction ratio, being in the range of more than 30 dB. Such high extinction ratio may be desirable in burst-mode systems based on Time Division Multiple Access (TDMA) technology allowing sharing the same fibre among many users transmitting in multipoint-to-point scheme without collisions. One example of such system is Passive Optical Network (PON) [1, 2] where TDMA is used for transmission in upstream direction from Optical Network Unit (ONU) to Optical Line Termination (OLT).

Biasing laser below its threshold is also possible when transmission system operates in continuos (CW) mode. It may be either deliberate (because of laser driver design) or may appear accidentally in conventional mean value stabilising loop when the ambient temperature range is exceeded.

Unfortunately, subthreshold biasing of the semiconductor laser gives rise to the turn-on delay what effectively shortens optical pulses. This is well-known phenomenon and occurs to some extent in all directly modulated semiconductor lasers operating in subthreshold bias regime [3, 4, 6]. Shortening of the pulses representing logical "ones" in obvious way limits the transmission rate of the fibre optic systems. Apart from that, it introduces jitter to the optical signal because amount of the delay is dependent on the number of consecutive "zero" symbols proceeding the symbol "one". This increases bit error rate (BER) of the optical link because the jittered signal can not be always sampled at the points of its maximum amplitude by periodic clock signal.

This paper focuses on the problem of BER degradation in fibre optic transmission systems exploiting laser transmitters operating with subthreshold bias. The method of analysis is proposed allowing calculating link BER that takes into account the turn-on delay jitter generated in the transmitting laser and assuming receiver noise to be an additive gaussian process. In addition, depending on transmission rate and modulation parameters, minimal values of biasing current are determined in terms of sensitivity penalty. Results of calculations are presented for the transmission rates ranging from 155 Mb/s up to 2.5 Gb/s. Model of the semiconductor laser used in the presented analysis was verified experimentally [7, 8, 9] and exploits only a few easy to obtain laser parameters.

2. TURN-ON DELAY JITTER

Operation of the semiconductor laser is based on the process of stimulated emission of photons. For this process to exist the concentration of electrical carriers in the laser active volume must reach the threshold level N_{TH}, what corresponds to some threshold current I_{TH}. If current flowing through the laser is below I_{TH} the emitted optical power comes from the spontaneous emission only and is quite small (may be as low as 0.1 μW for $0.2I_{TH}$ or a few μW for just at the threshold bias). To start the stimulated emission the concentration of the carriers in the laser active volume must first grow up to the threshold level thus optical response will be delayed with relation to the electrical current pulse (see Fig. 1a). The amount of this so-called turn-on delay depends on laser driving currents (initial bias I_0 and modulation amplitude I_1) and also on laser construction and semiconductor properties (like for example unimolecular, spontaneous and Auger recombination rates [4]).

When the carrier concentration once reaches the threshold level the laser starts to emit light. Their further response to the electrical current changes becomes very fast because of short photon lifetime (being typically in the order of a few ps [4, 6]). Thus, the decay of the laser light may often be assumed instantaneous after current decrease

below I_{TH}. It is important to notice however that the carrier concentration diminishes much slower and this phenomenon is the origin of turn-on delay jitter. When next current pulse is applied, the delay of optical response depends on the residual carrier concentration. This is sketched in Fig. 1b, where the rising edge of laser output delays more if the time when the laser is biased below threshold lasts longer. It is also seen that if the turn-off time is shortened, the residual carrier concentration stays significant and the turn-on delay is reduced.

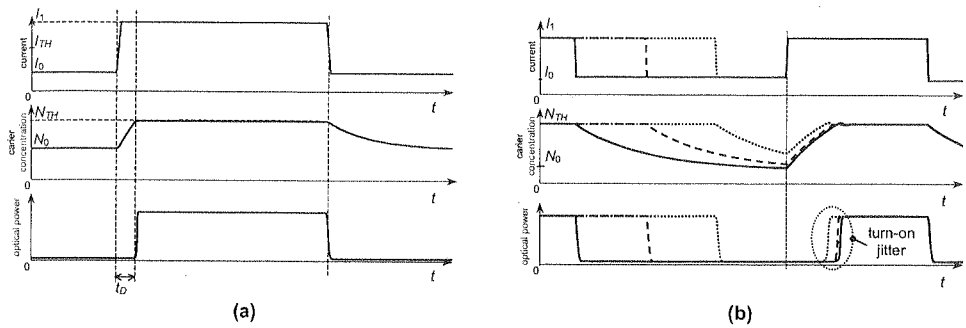


Fig. 1. Illustration of laser switching: turning on and off by current pulse (a), modulation by a digital waveform (b)

Rys. 1. Ilustracja procesu przełączania lasera: włączanie i wyłączanie (a), modulacja sygnałem cyfrowym (b)

The turn-on delay may be computed accurately using the set of laser rate equations [4]. Introducing the concept of Langevin noise sources allows including the effects of spontaneous emission noise adding some uncertainty to the turn-on delay [10, 11]. Such approach, however, requires not only the knowledge about the laser turn-off time and values of N_{TH}, I_0, I_1 but the complete set of parameters describing laser dynamics as well. Unfortunately, these parameters must be elaborately extracted from measurements for each particular laser [12, 13].

However, reducing our considerations to nowadays telecommunication-grade lasers operating in II and III fibre optic window it is justified to assume that nonradiative recombination is dominated by the Auger process with rate proportional to the cube of carrier concentration [4, 5]. Also, because of laser subthreshold biasing, the jitter contribution resulting from laser digital modulation may be assumed prevailing over the component caused by spontaneous emission noise [14, 15].

Taking above into account the laser dynamics for $N < N_{TH}$ may be approximated by simple differential equation [7]:

$$\frac{dN}{dt} = \frac{\eta I}{qV} - CN^3, \tag{1}$$

where C is Auger recombination coefficient, q is the elementary charge, V is the laser active volume, η denotes internal the quantum efficiency [4] and I is external laser driving current. After some algebraic operations [7, 9] equation (1) may be transformed into:

$$\frac{dJ}{dt} = \frac{1}{c} J^{2/3} (I - J), \tag{2}$$

where parameter c is equal to:

$$c = \frac{1}{3} \sqrt[3]{\frac{q^2 V^2}{\eta^2 C}}. \tag{3}$$

Variable J used in equation (2) is connected with the carrier concentration N by the relation:

$$J = N^3 \frac{qVC}{\eta}. \tag{4}$$

Straightforward integration of equation (2) gives the amount of time required to change equivalent current J in some boundaries. To obtain the turn-on delay calculation of the integral is required:

$$t_D = c \int_{J_I}^{I_{TH}} \frac{J^{-2/3}}{I_1 - J} dJ, \tag{5}$$

where the lower integral boundary J_I should be determined from the knowledge of the laser turn-off time. In case of digital transmission considered herein, the turn-off time is equivalent to the number of consecutive "zero" symbols transmitted before current "one". Thus if B designates the bit rate the value of J_I may be obtained solving equation (2) for times $t_n = n/B; n = 1, 2, \ldots$ with initial condition $J(0) = I_{TH}$.

Because of random nature of digital data transmitted in the optical link, the turn-on delay also displays random characteristics what may be described in terms of probability density function (pdf). Because laser turn-of time changes in a discrete fashion, turn-on delay pdf will be discrete as well. Assuming probabilities of sending logical symbols "zero" and "one" being the same and equal to $P_b = 0.5$, the probability that particular delay value occurs will decrease as 2^{-n} when the number of consecutive "zeros" increases. Example of such pdf drawn in semilogarythmic scale is shown in Fig. 2. It may be noticed that increments of turn-on delay shrink when the laser turn-off lasts longer. This allows considering only a few first delays as different probabilistic events and sum all remaining events into one. The number of distinguishable delays depends on laser parameters and biasing/modulating conditions, but in most cases it is enough to take into consideration up to 10 consecutive "zeros".

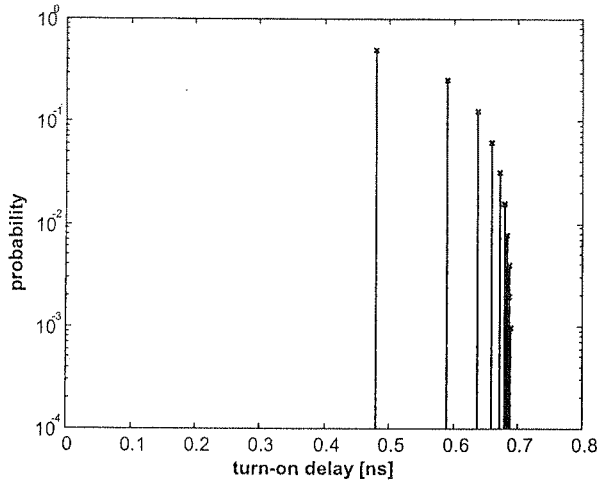


Fig. 2. An example pdf of turn-on delay for laser with I_{th} = 10 mA and c = 3 ns · mA$^{2/3}$, modulated by random digital signal from I_0 = 1 mA to I_1 = 20 mA with rate 622 Mb/s

Rys. 2. Przykład funkcji gęstości prawdopodobieństwa (pdf) czasu włączania lasera o parametrach I_{th} = 10 mA i c = 3 ns · mA$^{2/3}$, modulowanego losowym przebiegiem cyfrowym z prędkością 622 Mb/s

Model of the laser diode described above and used throughout further analysis was verified experimentally for a number of Multi Quantum Well (MQW) lasers [7, 8, 9]. Turn-on delay predictions calculated from equation (5) agreed with measured ones with accuracy of a few ps. Representative value of parameter c found from measurements is about 3 ns · mA$^{2/3}$. It was also verified that this value is quite independent on ambient temperature and biasing conditions so it is used next in all calculations [7].

3. MODEL OF THE TRANSMISSION LINK

In a high-speed fibre optic transmission link considered in the paper (see Fig. 3) digital data modulates laser source such that during logical "zero" laser is biased in the subthreshold regime (i.e. $I_0 < I_{TH}$) and during logical "one" current $I_1 > I_{TH}$ flows through the laser. For the purpose of further analysis we assume that the optical waveform produced by laser has zero rise and fall times. Modulated optical power is then sent down through the optical fibre to the receiver. After opto-electric (O/E) conversion and preamplification signal is next low-pass filtered (LPF) to minimise preamplifier noise. Filtered signal u is compared to the reference level U_{REF} and sampled by recovered serial clock in discrete time instants, i.e. signal detection is performed. Noise and signal distortions give rise to errors occurring in the detection process.

Filtering performed at the output of the preamplifier influences considerably the signal shape so characteristics of this filter should be deliberately chosen not to introduce

any substantial intersymbol interferences (ISI). Because it is not possible to realise Nyquist filter for gigabit-per-second rate signals, filter with fourth order Bessel-Thomson characteristics and 3 dB cut-off frequency equal to $0.75B$ is used very often in practical applications. It is also accepted to be a standard filter for SDH and SONET transport networks [16]. Such filter introduces only minor ISI to the digital signal if duration of each symbol is $1/B$ (see Fig. 4a) simultaneously limiting considerably noise bandwidth of the entire receiver.

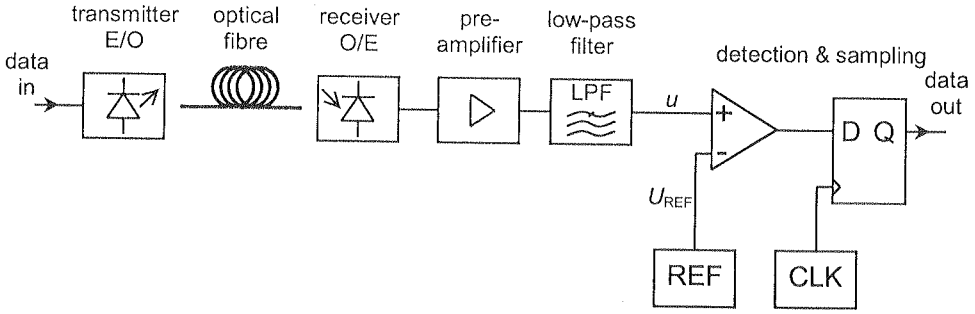


Fig. 3. Block diagram of the transmission link considered in the paper

Rys. 3. Schemat blokowy łącza transmisyjnego analizowanego w artykule

Because the laser operates in the subthreshold regime if "zero" is transmitted, the first pulse representing logical "one" is always shortened by turn-on delay. Thus, signal at the output of the fibre optic transmitter is nonlinearly distorted. Unfortunately filtering of such distorted signal by Bessel-Thomson LPF described above is not free from ISI. This is visible in Fig. 4b where examples of eye patterns for filtered undistorted and distorted signals are shown.

Because described phenomenon is nonlinear the degradation of the signal is not symmetric (see Fig. 4b). It affects signal in a twofold manner. The first one demonstrates equally in continuos and burst-mode transmission systems and results in unequal increase of elementary errors $P\{e_1\}$ and $P\{e_0\}$, where e_1 and e_0 denotes events of erroneous detection of "one" instead of "zero" and detection of "zero" instead of "one" respectively. As it is seen from Fig. 4b a broad plateau in the eye opening exists where the "low" value of signal is practically not affected by laser turn-on jitter. Comparing Fig. 4a and Fig. 4b it may be even noted that the interval where the signal takes its minimum lasts longer because of laser turn-on delay. Thus, $P\{e_1\}$ may be expected to change only slightly. On the other hand, sampling of the "high" value of the signal gives result substantially affected by laser turn-on jitter. It may be seen in the Fig. 4b that maximum value of the bit in the middle depends on symbols transmitted before and after it and is generally lower if "one" appears in the neighbourhood of "zeros". This results in increase of $P\{e_0\}$.

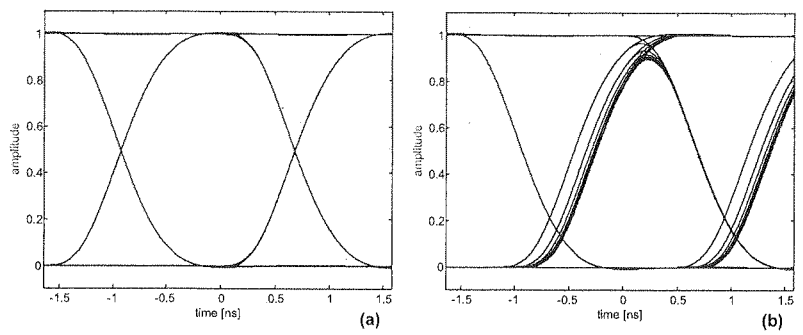


Fig. 4. Eye patterns at the output of fourth order Bessel-Thomson LPF: undistorted (a) and distorted by laser turn-on jitter (b). Parameters of modulation and laser are the same as for Fig. 2.

Rys. 4. Wykres "oka" sygnału na wyjściu filtru Bessla-Thomsona IV rzędu: niezniekształcony (a) i zniekształcony przez jitter czasu włączenia lasera. Parametry sterowania lasera takie same jak na Rys. 2.

The turn-on delay also decreases the mean value of the signal. This may be crucial in links transmitting data in continuous manner, where signal mean value is used as the reference level for signal detection in the receiver. Because of lowering the reference level some additional BER degradation may arise comparing to the case when this level is located exactly in the half of eye height. This point is not obvious however because lower reference level means more distance to the "high" level of the signal, which is corrupted by turn-on delay jitter. Thus in some cases increase in $P\{e_1\}$ may be overcompensated by decrease of $P\{e_0\}$.

4. ERROR PROBABILITY AND SENSITIVITY PENALTY

In digital transmission, the performance of the link is described in terms of BER. Its particular value depends on many factors, like for example signal to noise (SN) ratio, actual shape of the signal being compared, phase of the sampling clock and the reference level. If the transmitted signal were only attenuated minimum achievable BER would depend solely on noise generated in the receiver front-end. In case of signal considered herein, it is additionally distorted by other factors, i.e. turn-on delay and resulting ISI. If signal degradation were not too high, the performance of the link might be recovered if the link budget was increased by a proper amount. The increase required to restore BER to the same level, as it would be for undistorted signal is commonly termed as a sensitivity penalty.

The problem of sensitivity penalty due to turn-on delay jitter was previously analysed in the literature [17] using methodology described in [18]. The shortcoming of this approach lies in its very simple, raised-cosine model of signal pulses at the input of the decision circuit and modelling the turn-on delay jitter as continuous gaussian process.

Thus ISI generated in the receiver by filtering the distorted data signal are neglected and discrete nature of turn-on delay is lost. In the presented paper we undertake more rigorous approach, assuming realistic model for turn-on delay jitter and taking into account ISI generated by signal filtering as well.

Because of ISI presented in the signal, probability $P\{e\}$ of making an error in the detection of a particular symbol depends on some number of symbols transmitted before (precursors) and after (postcursors) this symbol [19]. Denoting by $\boldsymbol{S}$ a full set of different sequences s containing all symbols contributing to the amplitude of currently detected symbol (i.e. this particular symbol and all other interfering with it) the probability of error may be expressed in general as:

$$P\{e\} = \sum P\{e|s\}\,P\{s\}, \tag{6}$$

where $P\{s\}$ denotes the probability of occurrence of particular sequence s and $P\{e|s\}$ is the probability of making an error in the detection conditioned on the transmission of sequences. The summation in equation (6) should run through all sequences contained in the full set S.

For the purpose of further considerations we construct the set $\boldsymbol{S}$ taking into account limits put on the signal distortions by Bessel-Thomson LPF characteristics and turn-on delay jitter. First, we recognised that this is enough to limit the number of postcursors to one, because further symbols do not contribute to ISI. With the help of Fig. 1b and Fig. 4b it may be seen that ISI corrupts only symbol "one" and that in our case amount of ISI depends only on the number of precursors representing logical "zero". Thus, all symbols sent before "one" separated by a number of "zeros" from actually detected symbol "one" does not interfere with it. Additionally because of Bessel-Thomson filter characteristics a weak overshoot appears when changing symbol transmitted. Combining all these facts and assuming that only sequences with up to N "zero" symbols proceeding actually detected "one" give significant contributions to ISI we may state that the set $\boldsymbol{S}$ comprises the following events, grouped into eight categories:

i. exactly n "zeros" transmitted before sequence [1,0], where $n = 0 \ldots N$,
ii. more than N "zeros" transmitted before sequence [1,0],
iii. exactly n "zeros" transmitted before sequence [1,1], where $n = 0 \ldots N$,
iv. more than N "zeros" transmitted before sequence [1,1],
v. sequence [0,0,0] transmitted,
vi. sequence [0,0,1] transmitted,
vii. sequence [1,0,0] transmitted,
viii. sequence [1,0,1] transmitted.

In the above notation square brackets [ ] are used to denote the sequence of consecutive symbols transmitted. Events numbered i....iv. are relevant to detection of symbol "one" and the remaining ones to detection of symbol "zero". Although events v., vi. and vii., viii. are equivalent in the sense of signal produced at the LPF output at the

sampling instant, they were written as separate for notation clarity. This way we may think about the symbol being detected as the last but one symbol in the sequence $\boldsymbol{s}$.

Using the notation zeros(·) to designate a number of consecutive "zero" symbols we may rewrite equation (6) in the following way:

$$\begin{aligned} P_e = & \sum_{n=0}^{N} P\{e_1 \,|[1, zeros(n), 1, 0]\} \cdot P\{[1, zeros(n), 1, 0]\} + \\ & P\{e_1 \,|[0, zeros(N), 1, 0]\} \cdot P\{[1, zeros(N), 1, 0]\} + \\ & \sum_{n=0}^{N} P\{e_1 \,|[1, zeros(n), 1, 1]\} \cdot P\{[1, zeros(n), 1, 1]\} + \\ & P\{e_1 \,|[0, zeros(N), 1, 1]\} \cdot P\{[1, zeros(N), 1, 1]\} + \\ & P\{e_0 \,|[0, 0, 0]\} \cdot P\{[0, 0, 0]\} + \\ & P\{e_0 \,|[0, 0, 1]\} \cdot P\{[0, 0, 1]\} + \\ & P\{e_0 \,|[1, 0, 0]\} \cdot P\{[1, 0, 0]\} + \\ & P\{e_0 \,|[1, 0, 1]\} \cdot P\{[1, 0, 1]\} \end{aligned} \tag{7}$$

It should be pointed out that event of sending exactly n "zeros" before some sequence is equivalent to sending symbol "one" just before these n "zeros"; event of sending more than N "zeros" means sending one more "zero" at the beginning. These extra bits are included in the notation of equation (7).

Taking into account our previous assumption about equal probability of sending "zero" and "one" we may recognise that probabilities of particular events are equal:

$$P\{[1, zeros(n), 1, 0]\} = 2^{-n-3}, \tag{8}$$

$$P\{[0, zeros(N), 1, 0]\} = 2^{-N-3}, \tag{9}$$

$$P\{[1, zeros(n), 1, 1]\} = 2^{-n-3}, \tag{10}$$

$$P\{[0, zeros(N), 1, 1]\} = 2^{-N-3}, \tag{11}$$

$$P\{[0, 0, 0]\} = 2^{-3}, \tag{12}$$

$$P\{[0, 0, 1]\} = 2^{-3}, \tag{13}$$

$$P\{[1, 0, 0]\} = 2^{-3}, \tag{14}$$

$$P\{[1,0,1]\} = 2^{-3}. \tag{15}$$

It may be checked by inspection that the sum of probabilities (8)...(15) for all events defined in i....viii. above is equal to one.

Probabilities of error in the detection conditioned on the particular sequence transmitted depend on noise characteristics of the optical receiver front-end and the method of detection used. Assuming threshold detection and commonly accepted model of the noise in the form of additive, white gaussian (AWG) process the probabilities of error may be written in the form:

$$P\{e_1|s\} = \frac{1}{2}\mathrm{erfc}\left(\frac{u_1 - U_{\mathrm{REF}}}{\sqrt{2}\sigma_N}\right), \tag{16}$$

$$P\{e_0|s\} = \frac{1}{2}\mathrm{erfc}\left(\frac{U_{\mathrm{REF}} - u_0}{\sqrt{2}\sigma_N}\right), \tag{17}$$

where u_1 and u_0 are respectively amplitudes of received signal representing logical "one" and "zero" under detection, taking into account distortions from all precursors and postcursors making up sequence s, U_{REF} is the threshold level used for signal comparison and σ_N denotes standard deviation of the receiver noise. In particular values of u_1 and u_0 depend on the phase of sampling.

Having formula for BER, sensitivity penalty α_{td} induced by turn-on delay jitter may be calculated. It requires finding the appropriate value of σ_N from equation (7) assuming desired error probability $P\{e\}$ (e.g. 10^{-12}). Sensitivity penalty may be calculated accordingly to the formula:

$$\alpha_{td} = 20\log(\sigma_{N0}/\sigma_N), \tag{18}$$

where σ_{N0} is standard deviation of receiver noise giving the same desired error probability $P\{e\}$ under the assumption that no signal distortion occurs.

As it was previously noted, turn on delay influences the mean value of the signal as well. Because LP filtering does not change it, the mean value may be calculated directly from the signal at the laser output. Assuming that turn-on delays caused by N and $N+1$ consecutive "zeros" are indistinguishable the mean value may be obtained from the formula:

$$E\{u\} = \sum_{n=0}^{N}\left(1 - \frac{t_D(n)}{T}\right)P\{[1, zeros(n), 1]\} + \left(1 - \frac{t_D(N)}{T}\right)P\{[0, zeros(n), 1]\}, \tag{19}$$

where it was assumed that the signal is normalised to one at the LPF input and is equal to zero in the "low" state. Symbol $t_D(n)$ is used to designate laser turn-on delay after transmitting exactly n "zeros" before currently detected "one". Value of the delay may be calculated using equation (5). Probabilities of sequences in equation (19) are equal to:

$$P\{[1, zeros(n), 1]\} = 2^{-n-2}, \tag{20}$$

$$P\{[0, zeros(N), 1]\} = 2^{-N-2}. \tag{21}$$

In the Fig. 5 the dependence of $E\{u\}$ on the amount of subthreshold biasing is presented, drawn for different transmission rates and laser "high" current. It may be seen that the signal mean value increases almost linearly with increasing laser current in the "low" state. In many cases, however difference between 0.5 and actual mean value is quite significant. Generally, this difference can not be neglected in BER calculations if signal mean value is used as a reference level, especially for higher bit rates (≥ 622 Mb/s).

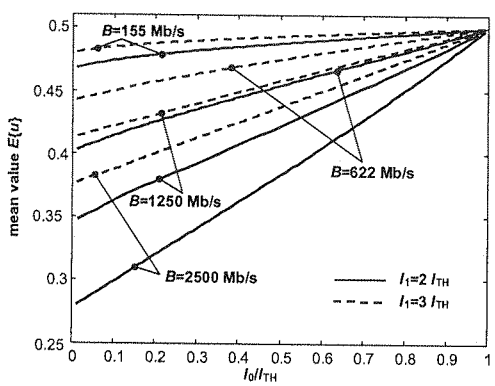


Fig. 5. Dependence of the mean value of the signal on subthreshold biasing for different data rates and modulation conditions. Assumed laser with parameter $c = 3$ ns · mA$^{2/3}$

Rys. 5. Zależność wartości średniej sygnału od stopnia polaryzacji podprogowej dla różnych warunków polaryzacji i prędkości transmisji. Przyjęto, że laser jest charakteryzowany przez parametr $c = 3$ ns · mA$^{2/3}$

5. RESULTS

Exploiting formulas and methods developed above a series of BER calculations was performed. Based on numbers of calculations it was observed that it is enough to

limit the summations in equations (7) and (19) up to $N = 10$ "zero" symbols preceding currently detected "one". Fourth order Bessel-Thomson frequency characteristics of the LPF with cut-off frequency equal to 0.75 B were assumed in the receiver. Because of this LP filtering all other signal distortions originating in laser (like e.g. relaxation oscillations [6]) are assumed being removed and are omitted in calculations.

In Fig. 6 graphs are presented showing dependence of BER on the I_0/I_{TH} ratio. Sampling phase was chosen optimal in the sense of minimum obtainable BER. Because acceptable BER for optical communication should be better than 10^{-9}, so the BER reference level (i.e. taking into account the receiver noise solely) equal to 10^{-12} was assumed. In Fig. 6a comparison of two different detection schemes is displayed. It may be noticed that using the threshold located at the half of the full bit amplitude in the detection (i.e. $U_{REF} = 0.5$) gives much better performance in general. The exception occurs when for higher bit rates (greater than 1Gb/s) laser is operated with small bias current I_0. In such cases comparing the signal with its mean value (i.e. $U_{REF} = E\{u\}$) results in lower BER.

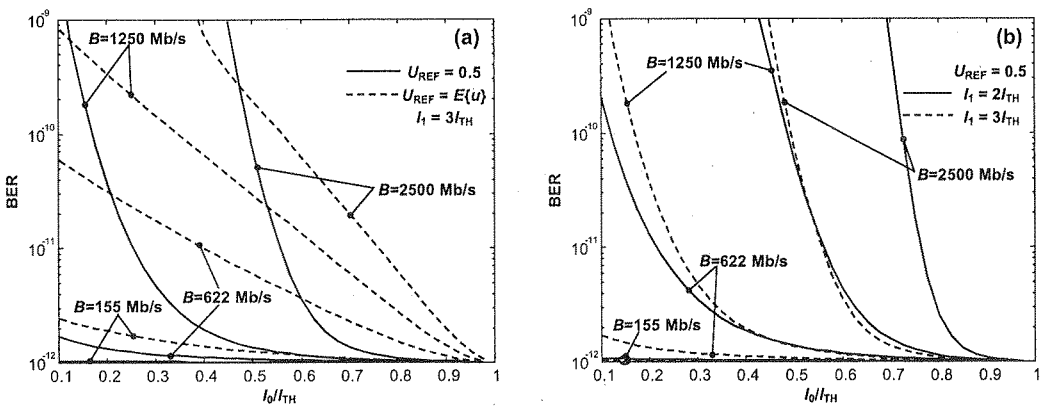


Fig. 6. BER versus I_0/I_{TH} ratio: for different detection reference levels (a), for different I_1 current values (b)

Rys. 6. BER w funkcji stosunku I_0/I_{TH}: dla różnych wartości progu decyzyjnego (a), dla różnych wartości prądu włączenia I_1 (b)

In Fig. 6b BER versus I_0/I_{TH} ratio is plotted for different transmission rates and I_1 currents, taking $U_{REF} = 0.5$. For the same B and I_0/I_{TH} ratio, more than an order of magnitude improvement in BER is observed when increasing laser current in the "high" state from $I_1 = 2I_{TH}$ to $I_1 = 3I_{TH}$. Similar relationship exists also if $U_{REF} = E\{u\}$.

In Fig. 7a locations of the optimum sampling instants (in the sense of minimum BER) are drawn for different transmission rates B, assuming the detection threshold $U_{REF} = 0.5$. It may be observed that the optimum sampling phase moves quite considerably when changing I_0/I_{TH} ratio, especially for higher bit rates. To estimate the influence of sampling phase on BER we performed calculations assuming sampling

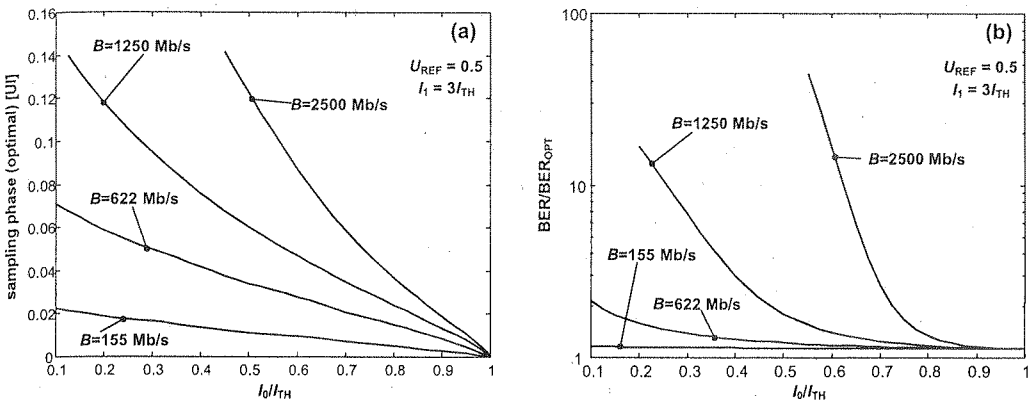


Fig. 7. Sampling phase giving minimum BER (a) and BER increase because of not optimal sampling (b)

Rys. 7. BER w funkcji stosunku I_0/I_{TH}: dla różnych wartości progu decyzyjnego (a), dla różnych wartości prądu włączenia I_1 (b)

instant being determined by the mean position of data edges (this corresponds to recovering the clock by phase locked loop (PLL)). In this case the position of the active edge depends on turn-on delay statistics (see Fig. 2). Resulting increase in the BER is presented in Fig. 7b. For data rates lower than 622 Mb/s this increase is practically unimportant, but, depending on I_0/I_{TH} ratio, it may become significant for higher data rates (it may grow to a few dozen for 2500 Mb/s).

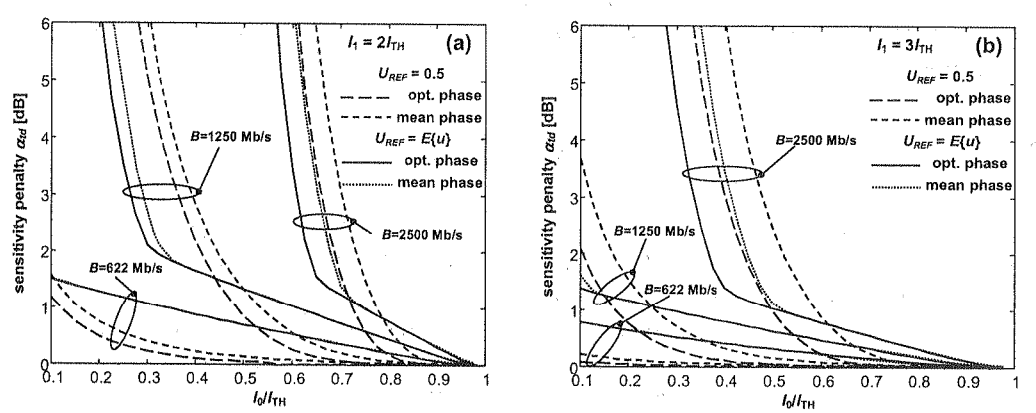


Fig. 8. Sensitivity penalty resulting from turn-on delay jitter: for $I_1 = 2I_{TH}$ (a) and $I_1 = 3I_{TH}$ (b)

Rys. 8. Strata czułości łącza spowodowana jitterem czasu włączenia lasera: dla $I_1 = 2I_{TH}$ (a) i dla $I_1 = 3I_{TH}$ (b)

To convert BER degradation into the amount of link budget decrease the sensitivity penalty α_{td} was next calculated accordingly to formula (18). Value of σ_N required to

get $P\{e\} = 10^{-12}$ was obtained by solving equation (7) numerically. Results are shown in Fig. 8 for different data rates and receiving schemes. Fig. 8a displays α_{td} when $I_1 = 2I_{TH}$ and Fig. 8b when $I_1 = 3I_{TH}$. In both figures curves drawn with dashed lines are for the reference level located in the half of full bit amplitude ($U_{REF} = 0.5$); curves drawn in solid and dots are for the reference level equal to the signal mean value ($U_{REF} = E\{u\}$). Labels "opt. phase" and "mean phase" are used in the figures to distinguish curves showing sensitivity penalty calculated for optimal sampling phase and phase taking into account turn-on delay statistics.

The essential result visible in Fig. 8 is that for each data rate a range of I_0/I_{TH} ratios exists where α_{PP} is relatively small and differs only slightly (a fraction of a dB) for all combinations of sampling phase and reference levels considered. Thus, turn-on jitter induced sensitivity penalty may be said to be relatively insensitive to detection strategy used in the receiver. Assuming that the acceptable sensitivity penalty is in the range of 3 dB it may be seen that laser operation with substantial subthreshold biasing is generally possible. For data rates in the range of 622 Mb/s, sensitivity penalty is lower than 2 dB for $I_0/I_{TH} < 0.1$ even if laser "high" current is $I_1 = 2I_{TH}$. Increase of modulating current to $I_1 = 3I_{TH}$ pushes data rate to about 1.25 Gb/s giving α_{td} still in the range of 3 dB for $I_0/I_{TH} < 0.1$. In this case laser modulation with 2.5 Gb/s rate is possible if $I_0/I_{TH} \geqslant 0.5$.

Sensitivity penalty was also calculated assuming second order filter characteristics with 3 dB bandwidth equal to 0.75 B. Obtained curves are very similar to those presented in Fig. 8 but are slightly shifted in the direction of smaller bias currents. It may be roughly stated that reducing slope of the filter characteristics allows decreasing I_0/I_{TH} ratio by about 0.05 to obtain α_{td} comparable to that for fourth order filter. In addition, because of different response shape of such filter the optimal sampling phase is shifted to the end of the bit duration.

6. CONCLUSIONS

Semiconductor laser biased in the subthreshold regime delays the rising edge of its optical response by the time dependent on the number of consecutive "zero" symbols transmitted. Thus the laser is the source of data dependent jitter. Filtering of the jittered data signal in the receiver produces some amount of ISI increasing BER of the transmission link. The problem becomes quite significant if the laser is biased substantially below threshold and weakly modulated.

In the paper problem of BER degradation because of turn-on delay jitter and resulting sensitivity penalty is investigated. The model allowing prediction of BER in fibre optic links with turn-on delay jitter is presented along with number of numerical results showing BER and sensitivity penalty dependence on laser modulation and biasing current under different detection strategies in the receiver.

The general conclusion that may be drawn is that negative consequences of turn-on delay jitter may be greatly reduced if the laser is strongly modulated. In practice laser

modulation is limited by its operating current I_{OP}, being typically about I_{TH} + 20 mA for nowadays transmission-grade lasers. Thus, this is highly desirable to have laser with large I_{OP}/I_{TH} ratio. This way MQW (multi-quantum well) lasers with I_{TH} lower than 10 mA are more suitable for subthreshold biasing than simple double heterostructure ones (having I_{TH} about twofold larger). Thus much better performance may be predicted for VCSELs (Vertical Cavity Surface Emitting Laser) with $I_{TH} \approx 2 \ldots 5$ mA.

It is also desirable to bias the laser as close to the threshold as possible. However the residual spontaneous emission in the subthreshold regime limits the allowable initial laser bias when high extinction is ratio required.

7. ACKNOWLEDGEMENTS

This work was supported by the Polish State Committee for Scientific Research (KBN) under the grant 4T11B05624.

8. REFERENCES

1. G. Kramer, G. Pesavento: *Ethernet passive optical network (EPON): building a next-generation optical access network.* IEEE Communications Magazine 2002, vol. 40, no. 2, pp. 66-73.
2. M. Yano, K. Yamaguchi, H. Yamashita: *Global Optical Access Systems Based on ATM-PON.* FUJITSU Sci. Tech. J. 1999, vol. 35, pp. 56-70.
3. D. Verhulst, Y.C. Yi, J. Bauwelinck, X.Z. Qiu, S. Verschuere, Z. Lou, J. Vandewege: *Theoretical and experimental study of laser turn-on delay in a GigaPON systems with pre-biasing bits.* Proceedings Symposium IEEE/LEOS Benelux Chapter, Amsterdam (Netherlands), 2002, pp. 290-293.
4. L.A. Coldren, S.W. Corzine: *Diode lasers and photonic integrated circuits*, New York, Wiley, 1995.
5. G. Morthier, P. Vankwikelberge: *Handbook of distributed feedback laser diodes*, Boston, Artech House, 1997.
6. B. Mroziewicz, M. Bugajski, W. Nakwaski: *Phisics of semiconductor lasers*, Warszawa, PWN, 1991.
7. P. Krehlik, Ł. Śliwczyński: *Modelling of dynamic performance of semiconductor lasers under subthreshold biasing.* Opto-Electron. Rev. 2004, vol. 12, no. 2, pp. 187-192.
8. P. Krehlik, Ł. Śliwczyński, A. Wolczko, M. Lipieński: *Zniekształcenia dynamiczne w laserowych układach nadawczych z podprogową polaryzacją wstepną.* Poznańskie Warsztaty Telekomunikacyjne, 2003, ss. 99-103.
9. Ł. Śliwczyński, P. Krehlik, M. Lipiński, A. Wolczko: *Modelowanie właściwości dynamicznych laserów MQW w warunkach polaryzacji podprogowej*, Poznańskie Warsztaty Telekomunikacyjne, 2003, ss. 225-230.
10. K. Obermann, S. Kindt, K. Petermann: *Turn-on jitter in zero-biased single mode semiconductor lasers*, IEEE Photon. Techn. Letters 1996, vol. 8, no. 1, pp. 31-33.
11. T. Czogalla, K. Petermann: *Turn-on jitter in zero-biased single mode vertical cavity surface emitting lasers*, LEOS Summer Topical Meeting, Montreal (Canada), 1997, pp. 55-56.

12. H.M. Salgado, M.M. Freire, J.J. O'Reilly: *Extraction of semiconductor intrinsic laser parameters by intermodulation distortion analysis.* IEEE Photon. Techn. Letters 1997, vol. 9, no. 10, pp. 1331-1333.
13. J.C. Cartledge, R.C. Srinivasan: *Extraction of DFB laser rate equation parameters for system simulation purposes.* IEEE J. Lightwave Technology 1997, vol. 15, no 5, pp. 852-860.
14. L. Pesquera, J. Revuelta, A. Valle, M.A. Rodriguez: *Theoretical calculation of turn-on delay time statistics of lasers under PRWM.* Proc. SPIE, 1997, vol. 2994, pp. 780-791.
15. S. Jamindar: *Performance evaluation of laser turn-on jitter in gigabit optical transmission systems. Third International Symposium of Communication Systems*, Networks and Digital Signal Processing, Staffordshire (England), 2002, paper F2.2.
16. ITU Recommendation G.957: Optical interfaces for equipments and systems relating to the Synchronous Digital Hierarchy. ITU, 1999.
17. T.M. Shen: *Timing jitter in semiconductor laser under pseudorandom word modulation.* Journal of Lightwave Technology 1989, vol. 7, no. 9, pp. 1394-1399.
18. G.P. Agrawal, T.M. Shen: *Power penalty due to decision-time jitter in optical communication systems.* Electronics Letters 1986, vol. 22, pp. 450-451.
19. J.G. Proakis: *Digital communications*, New York, McGraw-Hill, 1995.

Ł. ŚLIWCZYŃSKI, P. KREHLIK

WPŁYW OPÓŹNIENIA WŁĄCZENIA LASERA NA BITOWĄ STOPĘ BŁĘDU W ŁĄCZACH TELEKOMUNIKACYJNYCH

Streszczenie

Artykuł przedstawia analizę redukcję czułości, spowodowaną przez jitter powstający w bezpośrednio modulowanych laserach półprzewodnikowych, spolaryzowanych wstępnie prądem mniejszym niż prąd progowy. Wykorzystywany w analizie model lasera pozwala wyznaczyć opóźnienie włączenia lasera spolaryzowanego wstępnie poniżej progu w zależności od warunków sterowania. Zakłada on, że rekombinacja nośników przy takiej polaryzacji jest zdominowana przez proces Auger'a. Obliczenia stopy błędów łącza światłowodowego zostały wykonane dla prędkości transmisji w zakresie 155 Mb/s ÷ 2.5 Gb/s oraz dla różnych wartości stosunku prądu wstępnej polaryzacji do prądu progowego lasera. W analizie uwzględniono również zmianę wartości średniej sygnału, spowodowaną dynamicznym zniekształceniem wypełnienia przebiegu przez laser. Rezultaty otrzymane na podstawie przeprowadzonej analizy wskazują, że podprogowa polaryzacja lasera jest możliwa nawet dla prędkości transmisji rzędu 2.5 Gb/s jeśli prąd polaryzacji wstępnej jest utrzymywany w pobliżu połowy prądu progowego lasera. Natomiast dla prędkości transmisji poniżej 622 Mb/s nawet całkowite wygaszanie lasera w trakcie nadawania niskiej wartości sygnału wywołuje jedynie niewielką redukcję czułości rzędu 1.5 dB.

Słowa kluczowe: redukcja czułości, polaryzacja podprogowa lasera, opóźnienie włączenia lasera, TDMA, stopa błędów

Methods of converting of electric signals from photoelectric position transducer for discrimination of its motion direction and for increasing its measurement accuracy

ZBIGNIEW SZCZEŚNIAK, PH.D.

Kielce University of Technology, Faculty of Electrical Engineering, Automatic Control and Computer Science
Al. Tysiąclecia PP7, 25-314 Kielce, Poland
Z.Szczesniak@tu.kielce.pl

Otrzymano 2005.04.05
Autoryzowano 2005.06.23

In actual technical solutions two types of position transducers are distinguished: quantising transducers, called also incremental, and coding transducers. In the paper electronic methods of motion direction discrimination and position measurement for the incremental transducer are proposed. Original signals from the transducer are processed and transformed into the rectangular form [5]. The signals shaped in this way consist the input of the presented methods. First method is based only on logical functions of the transducer signals. The second one, besides the logical functions of the transducer signals, uses the motion pulses generated in RC circuits. The third of the methods is based on logical functions of the transducer signals and motion pulses generated in trigger circuits. It results from the analysis that systems forming one, two or four pulses for one transducer signal period, can be designed. This makes possible to increase additionally the accuracy of position measurement.

Key words: methods of motion direction discrimination, position measurement increasing of the photoelectric transducer accuracy

1. INTRODUCTION

In actual technical solutions two types of position transducers are distinguished: quantising transducers, called also incremental, and coding transducers. The subject of the analysis are output signals from the photoelectric incremental position transducer. Output signals from the photoelectric position transducer are two sinusoidal

signals, shifted by quarter period. The period of the signal is equal to the period of the transducer measurement bar grid scale. Electronic interpolating devices (frequency multiplying) and digitising devices (converting to the digital form) [5] convert those signals and in result two measurement rectangular signals, shifted by quarter period are obtained. Such signals give the possibility for elaboration of electronic methods of motion direction discrimination and position measurement, with the possibility of additional increment of the accuracy of the photoelectric transducer working with an object, what has been presented in the paper. Decreasing the quantisation level of the displacement of the drive mating with the transducer can be performed by more precise construction of the transducer or by more suitable conversion of the transducer signals [1], [2], [3], [4], [5], [6].

2. METHOD OF PHOTOELECTRIC TRANSDUCER MOTION DIRECTION DISCRIMINATION BASED ON LOGICAL FUNCTIONS OF THE TRANSDUCER SIGNALS

During the transducer operation electric rectangular signals A, G, shifted by quarter period are obtained on its two outputs (Fig. 1, Fig. 2). The task of the presented electronic system is counting the adequately generated motion pulses and distinguishing their sequence. The method is based on motion pulses generation on the base of the sum of the signals A + G and on the base of their product AG, what is presented in Fig. 1.

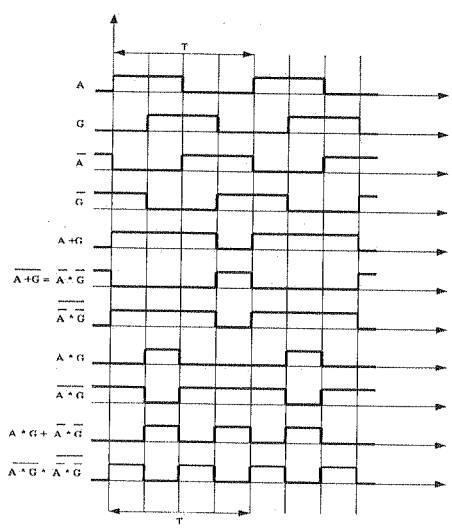


Fig. 1. Method of photoelectric transducer motion direction discrimination based on logical functions of the transducer signals

Rys. 1. Metoda rozróżniania kierunku ruchu optoelektronicznego przetwornika zrealizowana w oparciu o funkcje logiczne sygnałów tego przetwornika

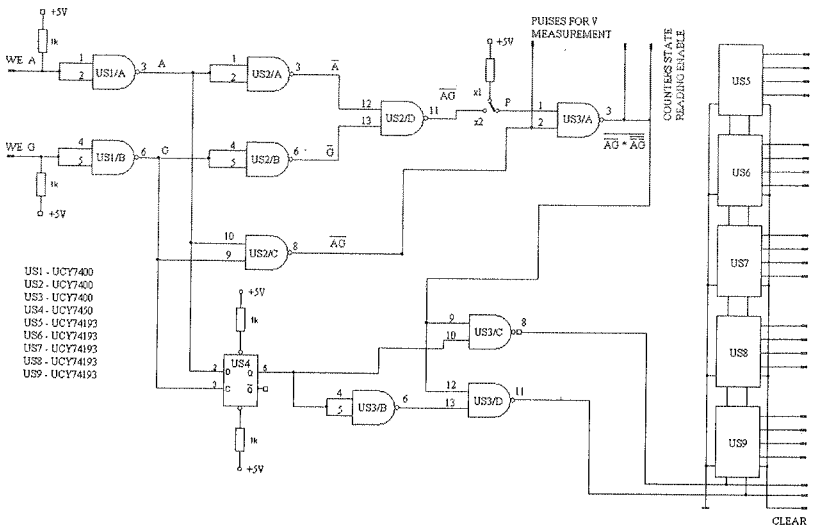


Fig. 2. System of photoelectric transducer motion direction discrimination based on logical functions of the transducer signals

Rys. 2. Układ rozróżniania kierunku ruchu optoelektronicznego przetwornika zrealizowany w oparciu o funkcje logiczne sygnałów

Making the sum of those pulses

$$WY = \overline{A + G} + AG \tag{1}$$

it is possible to generate, during the period T, 1 or 2 pulses, which are counted in the reverse order. The counting order is chosen dependently on the shift between the A and G signals, what is set by the trigger. The G signal is delivered to the clock input and the A signal is delivered to the D input of the trigger. Change of the G signal from "0" to "1", when A = "1", sets the Q trigger to "1". Counting to the right is then enabled. In case when change of the G signal appears at A = "0", i.e. when the G signal is leading to the A signal (change of the motion direction, the Q trigger changes its state and counting to the left is enabled. The system enables to choose the measurement accuracy x1 or x2, dependently on the setting of the P switch.

3. METHOD OF PHOTOELECTRIC TRANSDUCER MOTION DIRECTION DISCRIMINATION BASED ON LOGICAL FUNCTIONS OF THE TRANSDUCER SIGNALS AND MOTION PULSES GENERATED IN THE RC CIRCUITS

In this method (Fig. 3, Fig. 4.), by summing the suitable products of $A, \overline{A}, G, \overline{G}$ signals and signals arising from generation of pulses of duration τ (from their rising

edge) it is possible to count the pulses in the reverse counter, dependently on the transducer motion direction. When moving to the right the WY2 output of the NOR gate is set to "1", and on the WY1 output of the NOR gate a series of pulses repeated with the period of one transducer channel pulses is obtained:

$$WY1 = \overline{\overline{A} \cdot I\overline{G} + A \cdot IG + G \cdot I\overline{A} + \overline{G} \cdot IA}, \tag{2}$$

where: IA, IG – pulses from the A, G signal edge;
$I\overline{A}, I\overline{G}$ – pulses from the $\overline{A}, \overline{G}$ signal edge.

When moving to the left the WY1 output of the NOR gate is set to "1", and on the WY2 output of the NOR gate a series of pulses, similar to the ones when moving to the right are obtained:

$$WY2 = \overline{\overline{A} \cdot I\overline{G} + A \cdot IG + G \cdot I\overline{A} + \overline{G} \cdot IA}, \tag{3}$$

Those pulses are subtracted from the reverse counter state when moving to the right.

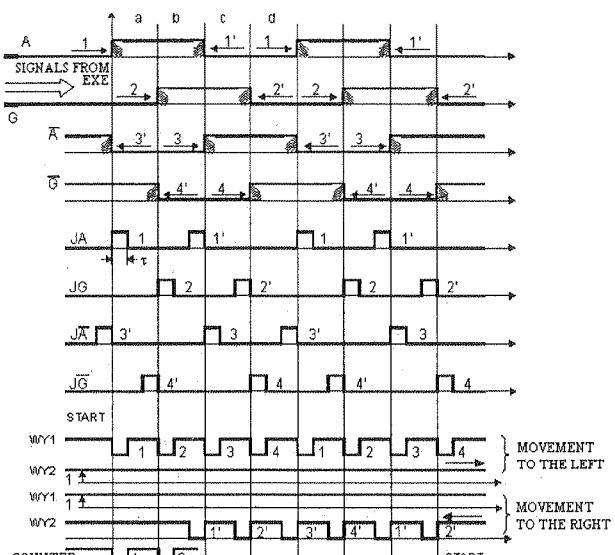


Fig. 3. Method of photoelectric transducer motion direction discrimination based on logical functions of the transducer signals and motion pulses generated in the RC circuits

Rys. 3. Metoda rozróżniania kierunku ruchu optoelektronicznego przetwornika zrealizowana w oparciu o funkcje logiczne sygnałów tego przetwornika i impulsów ruchu wygenerowanych w układach RC

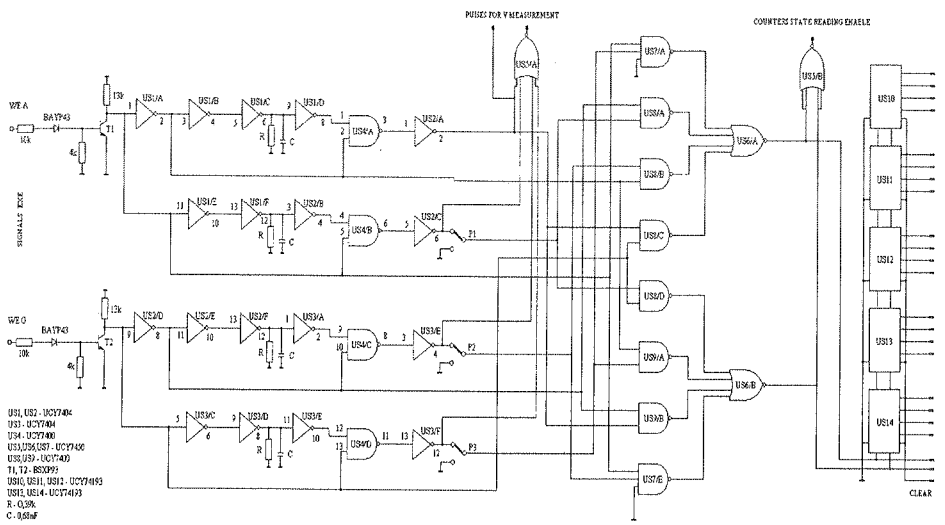


Fig. 4. System of the photoelectric transducer motion direction discrimination based on logical functions of the transducer signals and motion pulses generated in the RC circuits

Rys. 4. Układ rozróżniania kierunku ruchu optoelektronicznego przetwornika zrealizowany w oparciu o funkcje logiczne sygnałów tego przetwornika i impulsów ruchu wygenerowanych w układach RC

Change of the movement direction can appear in four different states a, b, c and d, what is depicted in Fig. 3. In every of those states the system must response identically, so the largest error of movement direction discrimination is equal to $\frac{1}{4}T$ (T – period of pulses in one transducer channel). It should be underlined that the system has the possibility of controlling the number of pulses counted for one period of the input signal (1, 2, 4), dependently on the settings of P1, P2, P3 switches (Fig. 4.). In this way there is a possibility to set the accuracy of system positioning on 1, 2 or 4 times.

4. METHOD OF PHOTOELECTRIC TRANSDUCER MOTION DIRECTION DISCRIMINATION BASED ON LOGICAL FUNCTIONS OF THE TRANSDUCER SIGNALS AND MOTION PULSES GENERATED IN THE TRIGGER CIRCUITS

Fig. 5 presents the system of direction discrimination and the way of output pulse forming in "+"IA and "-"IG channels. There are NAND gates with Schmitt triggers at the system input. Their task is to form steep slopes of input signals and to narrow the switching zone from 1 to 0 and inversely. It is important in case of possible object vibrations and of moving direction change. The pulse counted in the given direction must be counted again, in opposite movement direction, after having changed the moving direction.

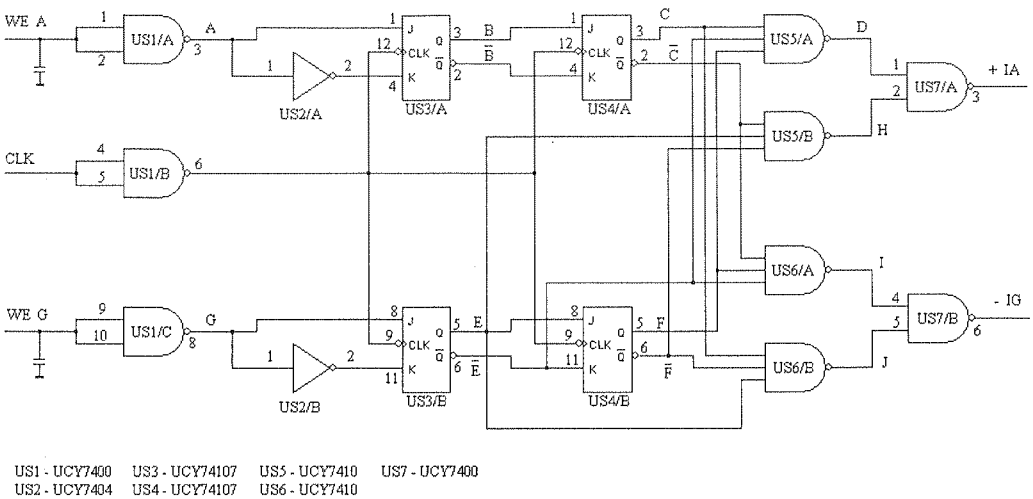


Fig. 5. System of photoelectric transducer motion direction discrimination based on logical functions of the transducer signals and motion pulses generated in the trigger circuits

Rys. 5. Układ rozróżniania kierunku ruchu optoelektronicznego przetwornika zrealizowany w oparciu o funkcje logiczne sygnałów tego przetwornika i impulsów ruchu wygenerowanych w układach przerzutnikowych

The Schmitt gates formed output pulses at A and G points (Fig. 5) have the polarity of rotary-pulse transducer output pulses because of the negations performed by the photoinsulator (optoelectronic coupler).

The system of JK flip-flops and gates perform basic functions of the movement direction discrimination. The JK flip-flops are synchronised with the CLK clock pulses the combinational circuit performs the following function:

$$\text{“+”IA} = (C\,\overline{E}\,F)\ \text{v}\ (\overline{C}\,E\,\overline{F}) \tag{4}$$

$$\text{“–”IG} = (\overline{C}\,\overline{E}\,F)\ \text{v}\ (C\,E\,\overline{F}) \tag{5}$$

The rule of output pulse forming has been presented in Fig. 6. Pulses for positive direction are obtained at A = “1” and the falling slope of the G signal and at A = “0” and the rising slope of the G signal. Pulses for the direction taken as negative are obtained at A = “1” and the rising slope of the G signal and at A = “0” and the falling slope of the G signal.

The system allows forming two output pulses in the given direction during one period of the input signal. The pulses are then counted in the pulse counters. The reverse counter state is the object position measure.

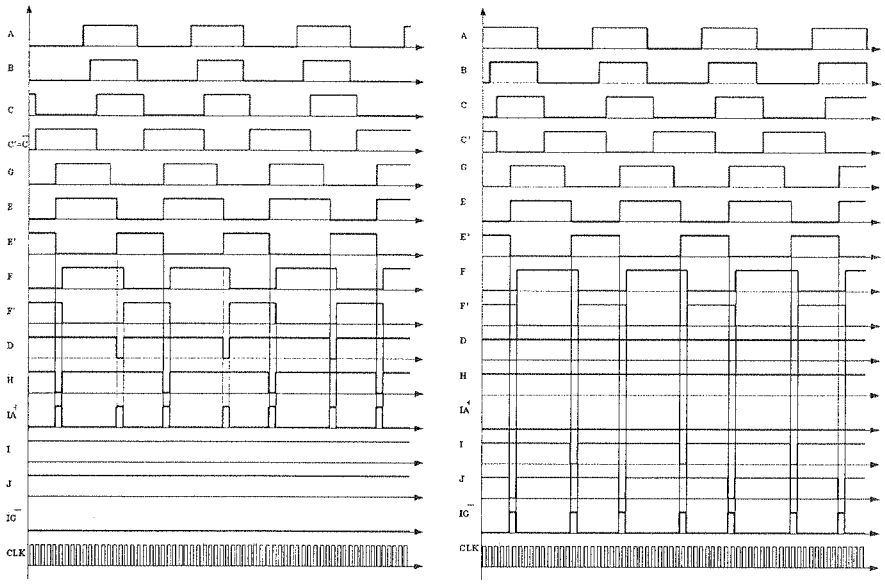


Fig. 6. Method of photoelectric transducer motion direction discrimination based on logical functions of the transducer signals and motion pulses generated in the trigger circuits for movement to the right and to the left

Rys. 6. Metoda rozróżniania kierunku ruchu optoelektronicznego przetwornika zrealizowana w oparciu o funkcje logiczne sygnałów tego przetwornika i impulsów ruchu wygenerowanych w układach przerzutnikowych dla ruchu w prawo i w lewo

5. CONCLUSIONS

1. Presented systems afford possibilities for the correct recognising the object position, independently on the motion direction and eliminate the incorrect recognition in case of object vibrations. Reverse counters, automatically giving the value determining the object position at their output can perform input pulse bi-directional counting.

2. It results from the presented methods of photoelectric transducer motion direction discrimination, that systems forming one, two or four pulses for one transducer signal period can be designed. The systems additionally increase the position measurement accuracy. It depends on the number of signal slopes analysed for one period.

6. REFERENCES

1. D. Hanselman: *Technigues for improving resolver-to-digital conversion accuracy.* IEEE Trans. on Idustrial Elekctr., Vol. 38, No. 6, 1991, pp. 501-504.
2. K. Holejko: *Precyzyjne elektroniczne pomiary odległości i kątów.* Warszawa, WNT 1981.

3. Zb. Szcześniak, K. Sikora, A. Pizoń, I. Smolewski: *Prototype construction of the electronic system for forging height measurement for the 30 MN press in Warsaw Steel Works and author supervising on construction and practical application of the system.* Elaborated in Cracow University of Technology for Warsaw Steel Works. Stage II, 1989 (in Polish).
4. Zb. Szcześniak, K. Sikora, A. Pizoń, T. Stefański: *KBN grant, "Microprocessor controlled hydraulic drives with continuously working valves".* Elaborated in Kielce University of Technology. Stage II, 1993 (in Polish).
5. Зб. Щесьняк, М. Дорожовець: *Метод помноження частоти вимирювального сигналу фотоелектричного перетворювача положення.* Висник НУ "Львивська Политехника", "Комп'ютерни мережи та системи". N523. С.154-157. 2005р.
6. Zb. Szcześniak: *A method of interpolation of photoelectric transducer electric signals in the position measurements.* Electronics no 1/2005 Warsaw 2005 pp. 74-76.

Zb. SZCZEŚNIAK

METODY PRZETWARZANIA SYGNAŁÓW ELEKTRYCZNYCH OPTOELEKTRONICZNYCH PRZETWORNIKÓW POŁOŻENIA W CELU WYRÓŻNIANIA JEGO KIERUNKU RUCHU ORAZ ZWIĘKSZENIA DOKŁADNOŚCI POMIARU

Streszczenie

W obecnie stosowanych rozwiązaniach technicznych wyróżnia się dwa rodzaje przetworników położenia: kwantujące, zwane także inkrementalnymi, oraz kodujące. W artykule zaproponowano elektroniczne metody rozróżniania kierunku ruchu i pomiaru położenia dla przetwornika inkrementalnego. Sygnałom oryginalnym otrzymywanym z wyjścia przetwornika nadaje się poprzez odpowiednie przetwarzanie kształt prostokątny [5]. Tak uformowane sygnały stanowią wejście przedstawionych metod. Pierwsza z nich bazuje na funkcjach logicznych sygnałów samego przetwornika. Druga metoda, obok funkcji logicznych sygnałów przetwornika, wykorzystuje impulsy ruchu wygenerowane w układach RC. Trzecia metoda oparta jest jednocześnie o funkcje logiczne sygnałów przetwornika oraz impulsy ruchu wygenerowane w układach przerzutnikowych. Z przeprowadzonej analizy wynika, że możliwa jest konstrukcja układów formujących jeden, dwa lub cztery impulsy na okres sygnału przetwornika. Umożliwia to dodatkowe zwiększenie dokładności pomiaru położenia.

Słowa kluczowe: metody rozróżniania kierunku ruchu, pomiar położenia zwiększenie dokładności przetwornika optoelektronicznego

KWARTALNIK ELEKTRONIKI I TELEKOMUNIKACJI, 2006, 52, z. 1, ss. 31–48

Non-uniform, non-time decimated filter banks based on recursive FIR structures

JAROSŁAW GRONCZYŃSKI, JANUSZ MROCZKA

Politechnika Wrocławska, Katedra Metrologii Elektronicznej i Fotonicznej,
ul. B. Prusa 53/55, 50-317 Wrocław
e-mail: jargron@yahoo.com

Otrzymano 2004.12.01
Autoryzowano 2005.05.09

This paper presents a solution of non-uniform filter bank without time-domain decimation. The lack of computational efficiency loss was achieved by using recursive FIR structures. In this case a computational burden is independent of the impulse response length of component filters. The construction of basic complementary filter pairs was based on Recursive Fourier Transform algorithm. To improve frequency parameters of the component filters, a triangular window was used and realized in a recursive manner. The way of signal reconstruction was presented. Proposed filter bank has a linear phase response.

Key words: filter bank; Recursive Fourier Transform; recursive FIR structure; aliasing; frequency sampling filters

1. INTRODUCTION

The multirate signal processing is commonly used in contemporary filter banks. It is essential in case of the lossy signal compression systems due to the requirements of reducing the bitrate of an output signal. However, if sampling frequencies of signals at the input and output of the system are the same, using multirate processing is not obligatory. Examples of such applications are: dynamic channel equalizers, graphic equalizers for audio, noise reductors. In these cases non-time decimated filters can be used, but they usually lead to computational inefficiency. The exceptions are the frequency sampling recursive FIR structures [1], that can be used to create a simple, uniform, non-time decimated filter bank or to estimate a signal spectrum. Their advantage is the lack of aliasing components in all output signals and a simple signal reconstruction – a synthesis bank is unnecessary.

In many practical applications non-uniform filter banks are more suitable due to the human perception properties. Such banks are commonly used to realize the Discrete Wavelet Transform (DWT). Unfortunately direct use of the DWT in the mentioned above applications results in uncompensated aliasing components at the output of the Inverse DWT block – processed signal output. The aliasing components will appear if the ratios of signal amplitudes at the outputs of the DWT block are changed. The reason is the time decimation at every processing stage and impossibility of creating the ideal half-band filter. In every signal at the DWT outputs there is an important aliasing component, but if these signals are not changed, aliasing components will be cancelled after IDWT. Already such a situation is equivalent to the lack of the signal filtration – the output signal is the same as the input one. In the decimated structures the aliasing problem is often resolved by dividing the input band at every filtration stage into two unequal subbands instead of two equal, and decimating only one of them instead of both [2]. A non-uniform filter bank can also be constructed from the recursive FIR filters, which is the subject of this paper. In this case aliasing problem does not exist at all.

Multiresolution is the property of a non-uniform filter bank. In time decimated structures it is achieved through the multistage filtration connected with the decimation at every stage. In the presented concept of a non-uniform filter bank based on the recursive FIR structures the same effect is obtained in a different way. The multistage filtration is also used, but instead of using decimation, impulse response lengths of filters at every filtration stage are changed. It is possible due to a particular property of the recursive FIRs, where their computational efficiency does not depend on their impulse response lengths.

2. LOWPASS RECURSIVE FIR SECTIONS

The basic part of the considered filter bank is the recursively realized lowpass FIR filter. Its structure can be directly obtained from the Recursive Fourier Transform algorithm [3] which realizes the equation:

$$y(m,k) = \sum_{n=0}^{N-1} x(m-n) \cdot e^{\frac{-j2k\pi n}{N}}. \tag{1}$$

The estimation of the zero frequency spectrum coefficient k and shifting an analysis window by one sampling interval is equal to the lowpass filtration without the time decimation:

$$y(m,0) = \sum_{n=0}^{N-1} x(m-n). \tag{2}$$

Z-transform of (2) shows that this operation can be realized in a recursive manner:

$$Y(z) = X(z) \cdot \sum_{n=0}^{N-1} z^{-n} = X(z) \cdot \frac{1 - z^{-N}}{1 - z^{-1}}. \tag{3}$$

The frequency response of the achieved lowpass recursive FIR filter is the response of a well-known rectangular window. This window is a prototype of the filter. Frequency properties of the rectangular window are poor. An improvement can be achieved by using the triangular window instead. A structure of a recursive estimation of a signal spectrum with this window was presented in [4]. In the time domain the triangular window can be considered as a convolution of two rectangular windows with the lengths equal to the half length of the triangular window. It means that the Z-transform of the triangular window is the squared transform of the rectangular window with the scaling factor equal to the half length of the triangular window:

$$\begin{aligned} Y(z) &= X(z) \cdot \frac{2}{N} \cdot z^{-1} \cdot \left(\frac{1 - z^{-N/2}}{1 - z^{-1}} \right)^2 = \\ &= X(z) \cdot \frac{2}{N} \cdot z^{-1} \cdot \frac{1 - 2z^{-N/2} + z^{-N}}{(1 - z^{-1})^2}. \end{aligned} \tag{4}$$

An additional delay by one sampling interval had to be taken into consideration in (4), because the first time sample of the periodic triangular window should be zero. To avoid amplification of the output signal, normalization through dividing the output values by the factor N – length of the filter impulse response is required. In case of the triangular window this factor should be reduced to N/2, due to 6 dB attenuation of this window for 0 Hz frequency. The transfer function of the lowpass structure with the rectangular window used as a prototype is then defined:

$$H_{LP(R)}(z) = \frac{1}{N} \cdot \frac{1 - z^{-N}}{1 - z^{-1}}. \tag{5}$$

Adequately, the case of the triangular window is described by the equation:

$$H_{LP(T)}(z) = \left(\frac{2}{N} \right)^2 \cdot z^{-1} \cdot \frac{1 - 2z^{-N/2} + z^{-N}}{(1 - z^{-1})^2}. \tag{6}$$

3. BASIC COMPLEMENTARY **LP-HP** FILTER PAIRS

The lack of decimation in the lowpass structure and the linear phase property simplifies the construction of the complementary highpass structure. It is enough to subtract the filtered lowpass signal from the fullband input signal. Equalization of the

time shift between the input and the lowpass filtered signal is required before this operation. Conversion of the Z transfer functions (5), (6) to the frequency domain leads to the equations:

$$H_{LP(R)}(\omega) = e^{-j\omega\cdot\left(\frac{N}{2}-\frac{1}{2}\right)} \cdot \frac{1}{N} \cdot \frac{\sin(\omega N/2)}{\sin(\omega/2)}, \tag{7}$$

$$H_{LP(T)}(\omega) = e^{-j\omega\cdot N/2} \cdot \left(\frac{2}{N} \cdot \frac{\sin(\omega N/4)}{\sin(\omega/2)}\right)^2. \tag{8}$$

If the delay of the input signal is equal to N/2-1/2 sampling intervals, the phase of this signal is the same as at the output of the lowpass filter described by (7) and based on the rectangular window. It means that the value of N must be odd (delay factor must be an integer). Unlikely, in case of the transfer function (8) – proper for the triangular window, a delay factor is N/2. It makes N even. The reason for that difference is zero value of the first sample of the periodic triangular window.

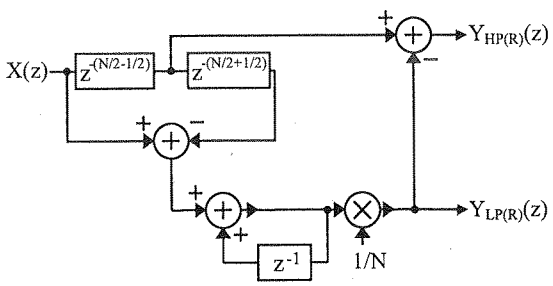


Fig. 1. Complementary LP-HP filter pair based on rectangular window

Rys. 1. Komplementarna para filtrów LP-HP oparta na oknie prostokątnym

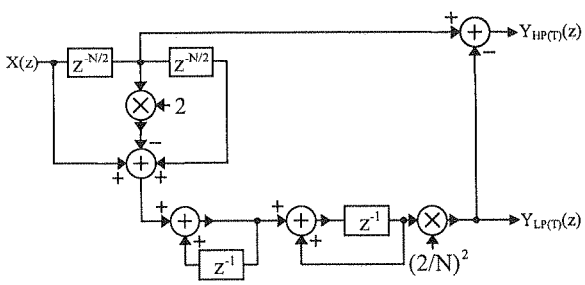


Fig. 2. Complementary LP-HP filter pair based on triangular window

Rys. 2. Komplementarna para filtrów LP-HP oparta na oknie trójkątnym

The structure of basic LP-HP filter pair obtained directly from the lowpass transfer function (5) is shown in Fig. 1. Adequately for the transfer function (6) it is Fig. 2. It is worth noticing that the second structure can be realized without multiplication units. If only N is the power of two, then all multiplications in the structure can be realized through a simple shift in binary arithmetic. Transfer functions of the highpass sections are defined by the equations:

$$H_{HP(R)}(\omega) = e^{-j\omega\cdot\left(\frac{N}{2}-\frac{1}{2}\right)} \cdot \left[1 - \frac{1}{N} \cdot \frac{\sin(\omega N/2)}{\sin(\omega/2)}\right], \tag{9}$$

$$H_{HP(T)}(\omega) = e^{-j\omega\cdot N/2} \cdot \left[1 - \left(\frac{2}{N} \cdot \frac{\sin(\omega N/4)}{\sin(\omega/2)}\right)^2\right]. \tag{10}$$

One can notice the only variable in the transfer equations (7), (8), (9), (10) is N – a number of impulse response samples of each filter pair. This value sets the frequency of the input band splitting. Figures 3 and 4 present magnitude responses of the lowpass and highpass sections, which were drawn for two values of N and for both structures of Fig. 1 and Fig. 2.

Another important property of the proposed structures is a constant amount of the arithmetic operations per one sampling interval regardless the N value. It is possible to divide the input band at different points without changing computational burden (even if lowpass characteristic is very narrowband). The N value influences only the length of the input delay line. This behaviour is completely different from the classical transversal FIR structures where the number of arithmetic operations is a direct function of the impulse response length.

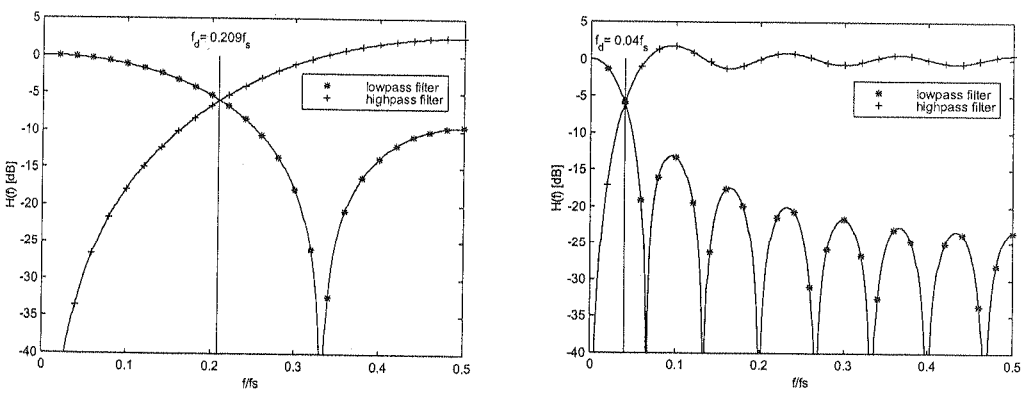


Fig. 3. Magnitude responses of the complementary LP-HP filter pair from Fig. 1 drawn for two values of N = 3 (left) and N = 15 (rigt)

Rys. 3. Charakterystyki amplitudowe komplementarnej pary filtrów z Rys. 1 wykreślone dla dwóch wartości N = 3 (po lewej) i N = 15 (po prawej)

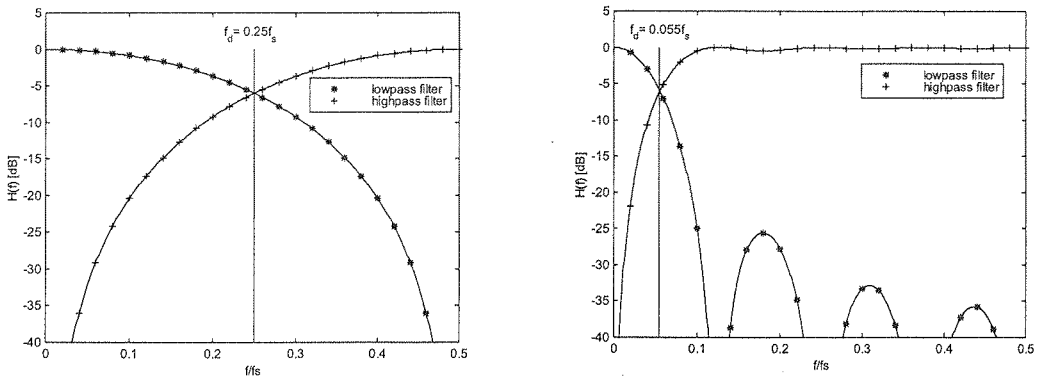


Fig. 4. Magnitude responses of the complementary LP-HP filter pair from Fig. 2 drawn for two values of N = 4 (left) and N = 16 (rigt)

Rys. 4. Charakterystyki amplitudowe komplementarnej pary filtrów z Rys. 2 wykreślone dla dwóch wartości N = 4 (po lewej) i N = 16 (po prawej)

Table 1 presents a number of arithmetic operations in both filter pairs. The structure based on the triangular window needs almost twice as many operations, but as it is shown in Fig. 3, 4, a better frequency response can be achieved – attenuation in the stopband of the lowpass section is higher, ripples in the passband of the highpass section are lower.

Table 1

Number of arithmetic operations per one sampling interval

Ilość operacji arytmetycznych na jeden okres próbkowania

Structure ⇒ ⇓ Operations	Fig. 1 (rect. window)	Fig. 2 (triang. window)
Additions	3	5
Multiplications	1	2

4. NON-UNIFORM, NON-DECIMATED FILTER BANK

A non-uniform and non-decimated filter bank is created through a cascade connection of M recursive LP-HP filter pairs. The way of connecting is shown in Fig. 5. It is important to notice that every LP-HP pair in this structure has a different impulse response length – N value. In case of the last pair (LP_M-HP_M) this value is the smallest and at every "lower" stage is consequently increased by a constant factor a in accordance to the equation:

$$N_{m-1} = \text{int}\{a \cdot N_m\}, \quad m \in N, \quad a > 1, \quad a \in R. \tag{11}$$

Finally LP_1-HP_1 pair at the first stage has the biggest impulse response length N_1. Each N_{m-1} value must be rounded to the nearest odd integer value in case of using LP -HP pair based on the rectangular window and to the even integer value in case of pairs based on the triangular window. Changing N means in fact, that every filter pair has a different split frequency between low and high subband (Fig. 3 and Fig. 4). In Table 2 a list of operations of the whole filter bank structure was presented. Due to the recursive realization a number of arithmetic operations only depends on the number of filtration stages M and chosen basic LP -HP structure (Fig. 1 or Fig. 2). The particular impulse response lengths N_m are not important.

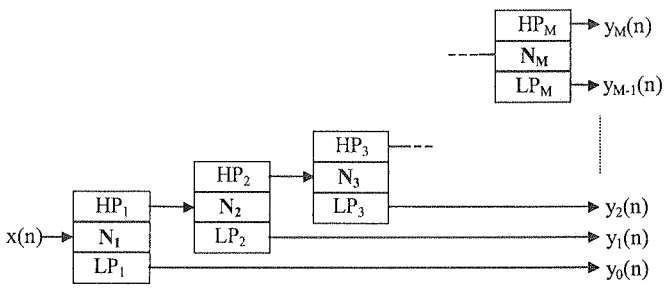


Fig. 5. Forming a non-uniform and non-decimated filter bank through a cascade connection of recursive LP -HP filter pairs

Rys. 5. Formowanie niejednorodnego i niedecymowanego banku filtrów przez kaskadowe łączenie par rekursywnych filtrów LP -HP

Table 2

Number of arithmetic operations per one sampling interval of filter bank consisted of M basic LP-HP filter pairs

Ilość operacji arytmetycznych na jeden okres próbkowania dla banku filtrów złożonego z M podstawowych par filtrów LP-HP

Used LP-HP pairs ⇒ ⇓ Operations	of Fig. 1 (rect. window)	of Fig. 2 (triang. window)
Additions	3M	5M
Multiplications	M	2M

A transfer function between the input and any passband output of the bank structure is expressed by the equation:

$$H_m(\omega) = \frac{Y_m(\omega)}{X(\omega)} = H_{LP(m+1)}(\omega) \cdot \prod_{l=1}^{m} H_{HPl}(\omega), \tag{12}$$

where $0 < m < M$.

Cases of lowpass and highpass outputs are defined as follows:

$$H_0(\omega) = \frac{Y_0(\omega)}{X(\omega)} = H_{LP1}(\omega), \tag{13}$$

$$H_M(\omega) = \frac{Y_M(\omega)}{X(\omega)} = \prod_{l=1}^{M} H_{HPl}(\omega). \tag{14}$$

Substitution of the LP and HP transfer functions (7), (9) to (12)÷(14) results in transfer functions of the filter bank consisted of the filter pairs using the rectangular window:

$$H_{0(R)}(\omega) = e^{\frac{-j\omega(N_1-1)}{2}} \cdot \frac{1}{N_1} \cdot \frac{\sin(\omega \cdot N_1/2)}{\sin(\omega/2)}, \tag{15}$$

$$H_{m(R)}(\omega) = e^{\frac{-j\omega\left(\sum_{l=1}^{m+1} N_l - m - 1\right)}{2}} \cdot \left[\frac{1}{N_{m+1}} \cdot \frac{\sin(\omega N_{m+1}/2)}{\sin(\omega/2)} \cdot \prod_{l=1}^{m}\left(1 - \frac{1}{N_l} \cdot \frac{\sin(\omega N_l/2)}{\sin(\omega/2)}\right)\right] \tag{16}$$

where $0 < m < M$,

$$H_{M(R)}(\omega) = e^{\frac{-j\omega\left(\sum_{l=1}^{M} N_l - M\right)}{2}} \cdot \prod_{l=1}^{M}\left(1 - \frac{1}{N_l} \cdot \frac{\sin(\omega \cdot N_l/2)}{\sin(\omega/2)}\right). \tag{17}$$

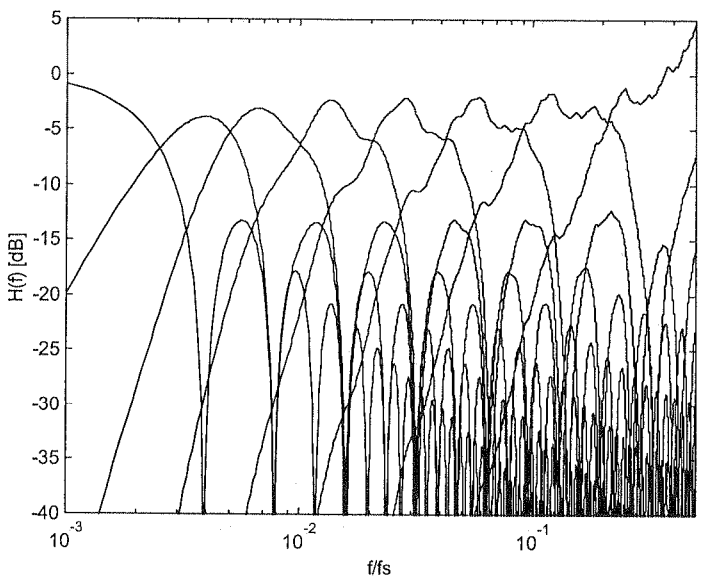


Fig. 6. Example magnitude responses of subbands of non-uniform filter bank based on the rectangular window. Next N values were assumed in seven filtration stages: [255, 127, 63, 31, 15, 7, 3]

Rys. 6. Przykładowe charakterystyki amplitudowe pasm niejednorodnego banku filtrów opartego na oknie prostokątnym. Przyjęto następujące wartości N w siedmioetapowej filtracji: [255, 127, 63, 31, 15, 7, 3]

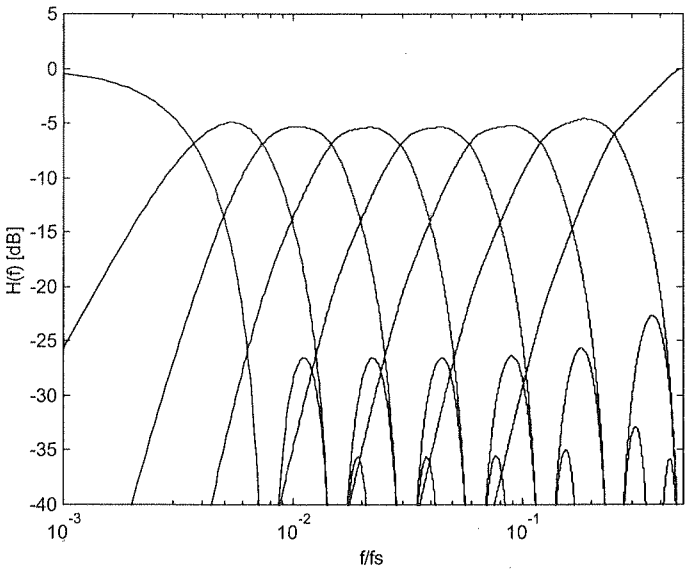


Fig. 7. Example magnitude responses of subbands of non-uniform filter bank based on the triangular window. Next N values were assumed in seven filtration stages: [256, 128, 64, 32, 16, 8, 4]

Rys. 7. Przykładowe charakterystyki amplitudowe pasm niejednorodnego banku filtrów opartego na oknie trójkątnym. Przyjęto następujące wartości N w siedmioetapowej filtracji: [256, 128, 64, 32, 16, 8, 4]

The example of magnitude responses were drawn in Figure 6. Due to the poor frequency parametrs of the rectangular window used as a prototype, properties of the constructed filter bank are not satisfactory. Strong ripples and unequality in higher subbands are disadvantageous. Also the increase of attenuation towards higher frequencies is slow in each subband. This behavior is the result of insufficient side lobes attenuation (13.2dB) and their low falling speed (–6dB/okt.) in the rectangular window. A better frequency response is achieved in a bank using the triangular window as a prototype of the component filter pairs. The triangular window has higher attenuation of the side lobes: 26dB and their falling speed: –12dB/okt.. Transfer functions between the input and outputs of the bank are then defined by equations:

$$H_{0(T)}(\omega) = e^{\frac{-j\omega N_1}{2}} \cdot \left(\frac{2}{N_1} \cdot \frac{\sin(\omega \cdot N_1/4)}{\sin(\omega/2)} \right)^2, \tag{18}$$

$$H_{m(T)}(\omega) = e^{\frac{-j\omega \sum_{l=1}^{m+1} N_l}{2}} \cdot \left[\left(\frac{2}{N_{m+1}} \cdot \frac{\sin(\omega \cdot N_{m+1}/4)}{\sin(\omega/2)} \right)^2 \cdot \prod_{l=1}^{m} \left(1 - \left(\frac{2}{N_l} \cdot \frac{\sin(\omega \cdot N_l/4)}{\sin(\omega/2)} \right)^2 \right) \right] \tag{19}$$

where $0 < m < M$,

$$H_{M(T)}(\omega) = e^{\frac{-j\omega \cdot \sum_{l=1}^{M} N_l}{2}} \cdot \prod_{l=1}^{M} \left(1 - \left(\frac{2}{N_l} \cdot \frac{\sin(\omega \cdot N_l/4)}{\sin(\omega/2)}\right)^2\right). \tag{20}$$

The example of magnitude characteristics achieved in this case were shown in Figure 7. To calculate them different values of constants were assumed: $N_M = 4$ and $a = 2$. For such values whole bank structure can be simply realized without a multiplication unit (bank consisted of filter pairs of Fig. 2). Widths of consecutive subbands then always increase two times and the final band division is the same as in the classical decimated octave filter bank, but sampling frequency is constant at every filter bank output.

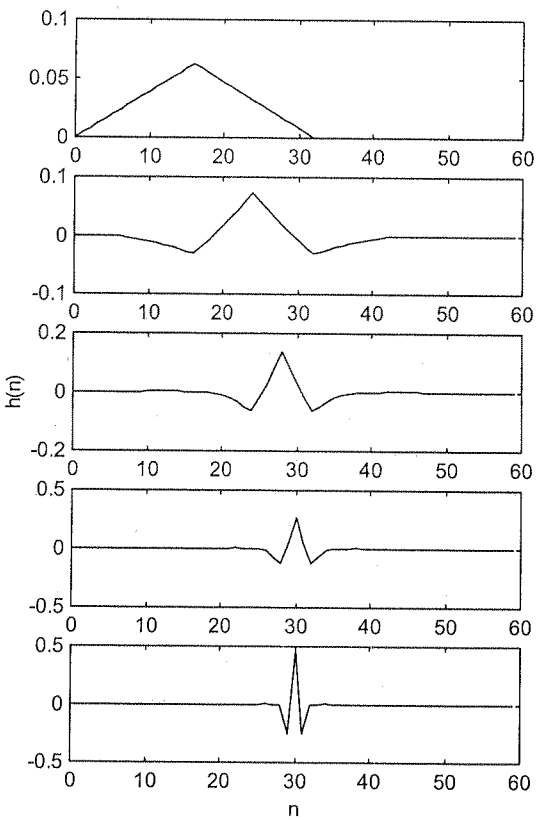


Fig. 8. Example impulse responses of non-uniform filter bank based on the triangular window (for lower frequency subbands at the top and higher at the bottom). Next N values were assumed in four filtration stages: [32, 16, 8, 4]

Rys. 8. Przykładowe odpowiedzi impulsowe niejednorodnego banku filtrów opartego na oknie trójkątnym (dla pasm o niższych częstotliwościach na górze, dla wyższych na dole). Przyjęto następujące wartości N w czteroetapowej filtracji: [32, 16, 8, 4]

Considering the common bank structure of Fig. 5 can lead to the conclusion that the product impulse responses corresponding to the subbands of higher frequencies are longer than for lower frequencies. Mathematically it is true, but presented in Fig. 8 results show clearly that in reality these proportions are completely different. Effective lengths of the impulse responses in the higher subbands are shorter than in the lower ones. In this case the statement of the effective length means the range where samples of the impulse response are important – their values are not equal to zero or near zero. Hence, real proportions between the lengths of the product impulse responses are similar to the achieved in the decimated non-uniform filter banks.

5. STABILITY CONDITION OF RECURSIVE **LP-HP** FILTER PAIRS

In theory FIR filters are always stable due to the lack of poles in the transfer function, but in case of the recursive realization stability conditions must be taken into account. The reason is the way of obtaining the finite impulse response – through cancellation of zeros with poles in a transfer function [1]. In most cases arithmetic operations realized in the finite length binary words are not ideal due to the round errors, so that cancellation may not also be ideal. Lack of full cancellation of the poles means that the recursive part of the structure may be unstable – dependently on the poles location on the Z plane. If a structure falls into an unstable state it becomes an IIR filter in fact. One can check that the poles of the recursive parts of the presented in Fig. 1 and Fig. 2 structures are placed on a unit circle in the Z-plane. Arithmetic errors can displace them outside the unit circle, that can be the reason of instability. Considerations of the fixed point arithmetic operations accuracy leads to the conclusion that there are some operations which can be done without any round errors. These operations are: addition, subtraction and multiplication without a right shift. All important parts of the algorithms in Fig. 1 and Fig. 2: taking signal from the input delay line and processing in the recursive loops, can be composed from the mentioned above operations. In the fixed point errorless arithmetic these algorithms are stable. Non-errorless are only scaling multiplications by factor 1/N and $(2/N)^2$ at the outputs of the recursive loops, but they do not influence stability. The location of the transfer function zeros and poles does not depend on them (in accordance to equations (5) and (6)).

6. SIGNAL RECONSTRUCTION

Due to the constant sampling frequency of signals at all outputs of the filter bank, signal reconstruction is easy and requires only equalization of the time delays between signals and summing them up then. Arguments of the transfer function (15)÷(20) show that the phase in case of each output is different. It means that delays are also different. Maximum delay is achieved for the last outputs of the filter bank – signals $y_M(n)$ and

$y_{M-1}(n)$ in Fig. 5. To equalize these differences delaying of other signals is required. In case of using the rectangular window (filter bank built on structures of Fig. 1) delay factors are defined by equations:

$$d_{M(R)} = d_{M-1(R)} = 0, \tag{21}$$

$$d_{m(R)} = \frac{\sum_{l=m+2}^{M} N_l - M + m + 1}{2}, \tag{22}$$

where $0 \leqslant m < M-1$.
Adequately, equations for the filter bank using the triangular window are:

$$d_{M(T)} = d_{M-1(T)} = 0, \tag{23}$$

$$d_{m(T)} = \frac{1}{2} \cdot \sum_{l=m+2}^{M} N_l, \tag{24}$$

where $0 \leqslant m < M-1$.

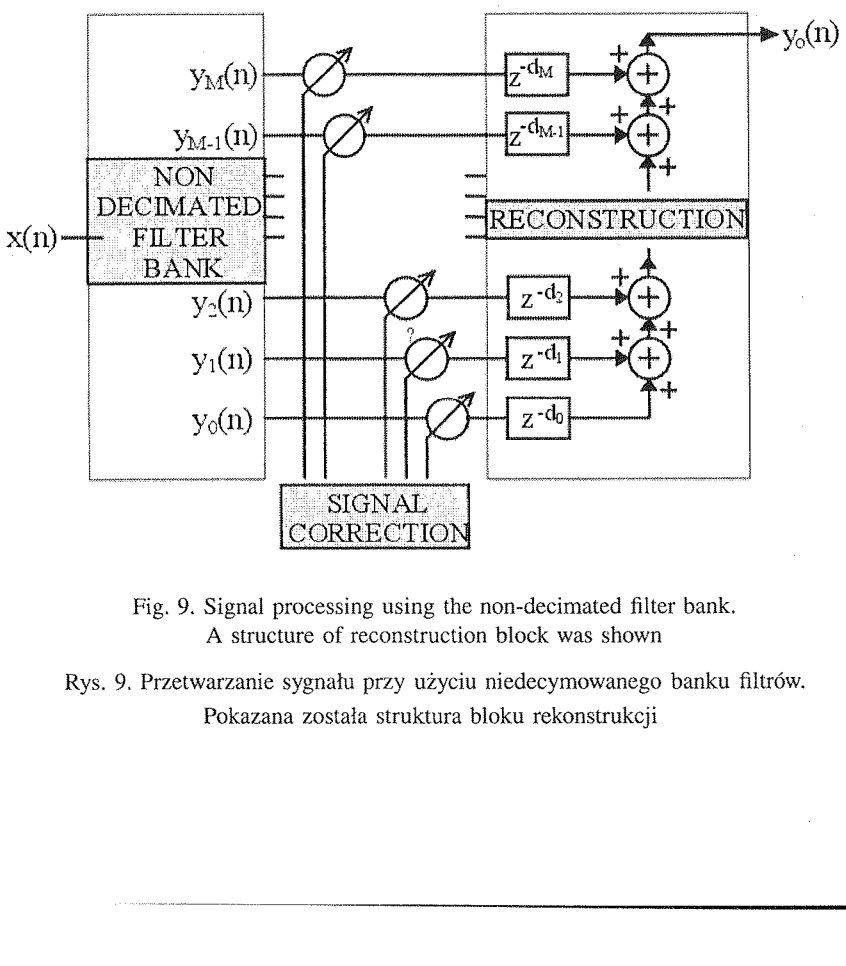


Fig. 9. Signal processing using the non-decimated filter bank. A structure of reconstruction block was shown

Rys. 9. Przetwarzanie sygnału przy użyciu niedecymowanego banku filtrów. Pokazana została struktura bloku rekonstrukcji

The way of signal reconstruction was presented in Figure 9. Factors defined in (21)÷(24) were used in delay blocks located in a reconstruction block in the picture. Apart from the reconstruction method, Fig. 9 shows also the way of signal processing by the presented in this paper filter bank. Amplitudes of signals at the outputs of the bank can be freely changed without introducing any aliasing components into output signal $y_{out}(n)$. It means that the bank can be easily adopted to realize shaping of the frequency response. This operation can be done dynamically, so it can be useful for the adaptive systems.

7. INCREASING FILTER BANK RESOLUTION

Frequency response of the filter bank depends on the frequency responses of the component filter pairs. Hence, any improvement of the frequency parameters of the bank needs improvement of parameters of the filter pairs first. In case of the increasing filter bank resolution it is required to reduce the width of the transition band in the magnitude responses of filter pairs. It can be done by changing a prototype function. Instead of the triangle window a better function that assures a narrower transition band and a magnitude response closer to the rectangle filter can be searched. However, finding such function is a hard task, because it has to fulfill many conditions: it must be realized in a recursive manner, generated filter structure must be simple to assure high computation efficiency, poles of the recursive part of this structure cannot be placed outside the unit circle in the Z-plane. Additionally, the function has to be symmetric to fulfill the linear phase condition. Much easier solution is the development of the structure in Fig. 2 (the structure in Fig. 1 will not be considered here, because it does not assure a satisfactory frequency response). Lowpass recursive FIR structures used to create LP-HP filter pairs were the frequency sampling filters. They were tuned to frequency 0Hz. Creating a lowpass section from more such filters and tuning them to different frequencies open new opportunities for shaping a frequency response of LP-HP pairs. In Figure 10 a structure was presented where a lowpass section was constructed from two frequency sampling recursive filters. They use the same input delay line, but in the second one an amplitude modulation with a complex exponential function was applied to shifting the signal spectrum [5]. In this way the added branch filters out a different part of the input spectrum from the basic branch (tuned to 0Hz). A signal at the output of the branch with the spectrum shift is complex, but its real part is the only important. The imaginary part has been discarded, so it does not have to be computed. Finally, signals from both branches are added. The sum is a lowpass filtered signal.

In literature another solution of the frequency sampling filters [1] (recursive estimation of signal spectrum: [4], [6]) is often presented. It was derived directly from the Recursive Fourier Transform algorithm and uses multiplications inside recursive loops instead of the complex modulators outside them. In theory the achieved results in both structures are the same, but in case of existing round errors, multiplications in

recursive loops can lead to filter instability. Round errors can arise in each iteration and can be increased in consecutive iterations due to next round operations. These structures do not have a forgetting factor – poles are placed on the unit circuit in the Z-plane. Using the structure that was presented in Fig. 10, multiplications in the input complex modulator can be realized without any right scaling shift, hence without round error. Instead, in the output modulator the scaling shift of the result is twice as long. Therefore, arithmetic operations inside the recursive loops and between the loops and the input delay line are still errorless.

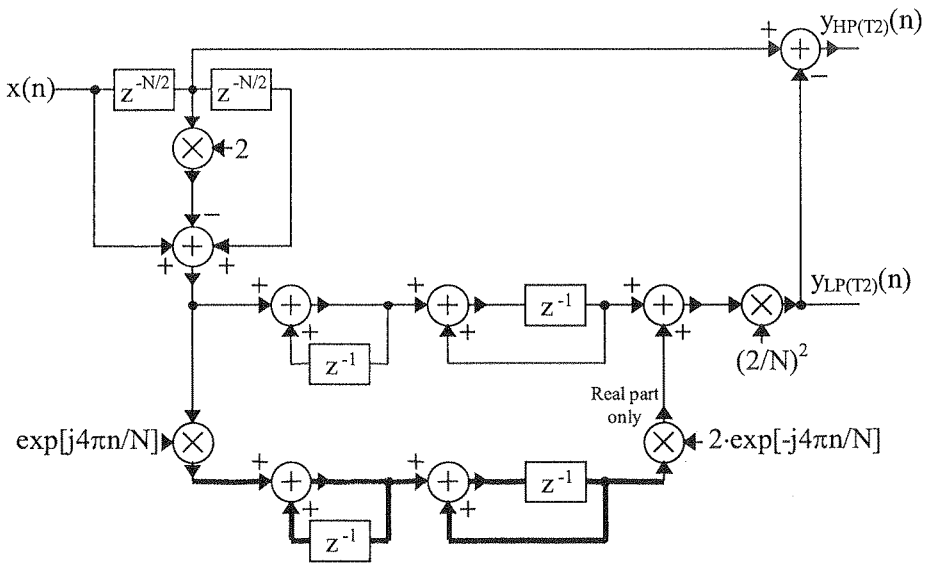


Fig. 10. Complementary LP-HP filter pair using two recursive branches to reduce the width of transition band (complex signal lines were thickened)

Rys. 10. Komplementarna para filtrów LP-HP wykorzystująca dwie gałęzie rekursywne w celu redukcji szerokości pasma przejściowego (linie sygnałów zespolonych zostały pogrubione)

The transfer function of the lowpass section with two recursive branches is expressed by equation:

$$H_{LP(T2)}(\omega) = e^{\frac{-j\omega N}{2}} \cdot \left(\frac{2}{N}\right)^2 \cdot \left[\left(\frac{\sin(\omega N/4)}{\sin(\omega/2 - 2\pi/N)}\right)^2 + \left(\frac{\sin(\omega N/4)}{\sin(\omega/2)}\right)^2 + \left(\frac{\sin(\omega N/4)}{\sin(\omega/2 + 2\pi/N)}\right)^2\right]. \quad (25)$$

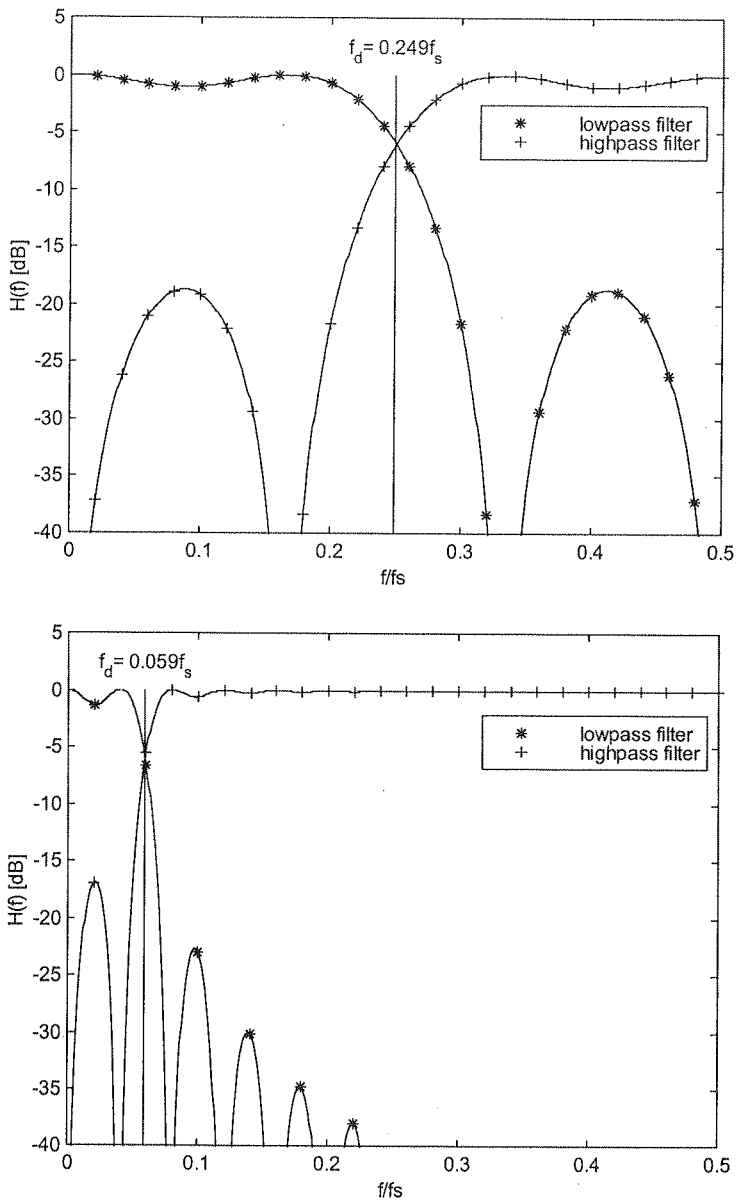


Fig. 11. Magnitude responses of the complementary LP-HP filter pair from Fig. 10 that were drawn for two values of N=12 (up) and N=50 (down)

Rys. 11. Charakterystyki amplitudowe komplementarnej pary LP-HP z rys. 10, które wykreślono dla dwóch wartości: N=12 (na górze) i N=50 (na dole)

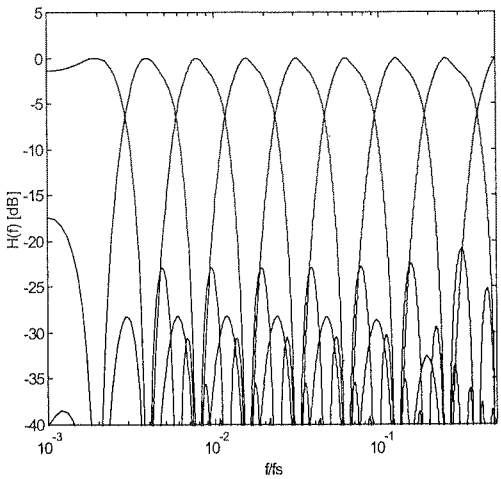


Fig. 12. Example magnitude responses of subbands of non-uniform octave filter bank using LP-HP filter pairs of Fig. 10. Next N values were assumed in 8 filtration stages: [1024, 512, 256, 128, 64, 32, 16, 8]

Rys. 12. Przykładowe charakterystyki amplitudowe pasm niejednorodnego oktawowego banku filtrów wykorzystującego pary filtrów z rys. 10. Przyjęto następujące wartości N w ośmioetapowej filtracji: [1024, 512, 256, 128, 64, 32, 16, 8]

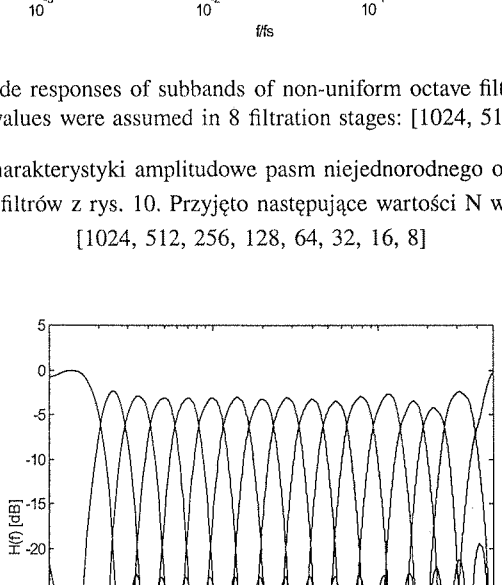


Fig. 13. Example magnitude responses of subbands of non-uniform half octave filter bank using LP-HP filter pairs of Fig. 10. Next N values were assumed in 16 filtration stages: [1448, 1024, 724, 512, 362, 256, 180, 128, 90, 64, 46, 32, 22, 16, 12, 8]

Rys. 13. Przykładowe charakterystyki amplitudowe pasm niejednorodnego półoktawowego banku filtrów wykorzystującego pary filtrów LP-HP z rysunku 10. Przyjęto następujące wartości N w 16-etapowej filtracji: [1448, 1024, 724, 512, 362, 256, 180, 90, 64, 46, 32, 22, 16, 12,8]

Figure 11 presents example magnitude responses of the lowpass and highpass sections of the structure of Fig. 10. Impulse response lengths Nm were chosen in such a way that the band division frequency is similar to the achieved in Fig. 4, where a lowpass section with one branch was used. As it was expected the magnitude responses became closer to rectangular and the widths of the transition bands were reduced. However, the cost of these improvements are additional arithmetic operations that must be done due to adding the second branch in the lowpass section: two multiplications of a real value by a complex value, two additions of complex values and one addition of real values. In this case a multiplication unit is required. Multiplication constants are not power of two, so they cannot be realized through a simple shift of a binary value. Moreover N_m values have to be lengthen to obtain the same band division frequencies, fortunately it does not influence the number of arithmetic operations. In Figures 12 and 13 magnitude responses of the improved filter bank were presented. In the first case the change of the subband width was preserved as in Fig. 7 – an octave bank (constant a in (11) equal 2). In the second case resolution of the filter bank was increased twice – a half octave bank. Thus constant a in (11) and the change of subband width had to be reduced to $\sqrt{2}$. Figure 13 shows that some problems with keeping this value in higher subbands exist, but generally the goal was achieved. The reasons are relatively big differences between rational values of N_m achieved directly from the multiplication by $\sqrt{2}$ and after rounding them to the nearest even values, if N_m are small. It concerns only higher subbands, because they are processed by the filter pairs with shorter impulse response lengths N_m.

8. CONCLUSION

In the paper the idea of construction of a non-uniform filter bank using recursive FIR filters was presented. Well-known window functions: the rectangular and the triangular were used as filter prototypes. These functions assured simplicity of the filter structures and their high computational efficiency. In case of an octave filter bank, the structure based on the triangular window makes this task possible to realize without a multiplication unit. However, frequency parameters of these windows can be in some applications insufficient. First of all it concerns the stopband attenuation. The value of this parameter directly depends on the side lobe level of the used window function. The attenuation in a stopband can be increased by choosing another window function. Trigonometric windows such as Hann, Hamming, Blackman and Blackman-Harris can be adopted here, but it is important to notice that the use of these windows does not have to lead to the computationally effective structures. Moreover some well-known window functions, such as for example Kaiser-Bessel windows cannot be realized in a recursive manner at all. It seems that the best solution to increase the stopband attenuation could be designing a special function to this application, but it is a hard task because many additional conditions must be fulfilled, especially a stability condition. Increasing the frequency resolution of the bank is much simpler task. In the paper a half octave filter

bank was presented as an example. The key to the increase of resolution is reducing the widths of component filters transition bands. It can be done through increasing the number of recursive branches used to the construction of basic LP-HP pairs. However, this solution leads to the increase of the number of arithmetic operations, so designing a new prototype function could be a better solution also in this case. It is the subject of further research.

9. REFERENCES

1. J.G. Proakis, D. G. Manolakis: *Digital Signal Processing – Principles, Algorithms, and Applications*, Macmillan 1988, p. 472, p. 285.
2. N. J. Fliege: *Multirate Digital Signal Processing*, Wiley 1999, p. 244.
3. J. H. Halberstein: *Recursive, Complex Fourier Analysis for Real-Time Applications*, Proc. IEEE, vol. 54, 1966, p. 903.
4. R. Hartley, K. Welles: *Recursive Computation of the Fourier Transform*, IEEE International Symposium on Circuits and Systems 1990, vol. 3, New York, pp. 1792-1795.
5. A. V. Oppenheim, A. S. Willsky: *Signals and Systems*, Prentice-Hall 1983, p. 460.
6. J. H. Kim, T. G. Chang: *Analytic Derivation of the Finite Wordlength Effect of the Twiddle Factors in Recursive Implementation of the Sliding-DFT*, IEEE Trans. Signal Processing vol. 48, May 2000, pp. 1485-88.

J. GRONCZYŃSKI, J. MROCZKA

NIEJEDNORODNE, NIEDECYMOWANE CZASOWO BANKI FILTRÓW OPARTE NA REKURSYWNYCH STRUKTURACH FIR

Streszczenie

Artykuł porusza zagadnienia cyfrowej filtracji i przetwarzania sygnałów. Przedstawiono w nim niejednorodny bank filtrów zrealizowany bez czasowej decymacji sygnału. W obecnych konstrukcjach banków niejednorodnych decymacja jest kluczowym elementem, który umożliwia uzyskiwanie wysokiej efektywności obliczeniowej. Jednak w przypadku modyfikacji stosunków amplitud składowych częstotliwościowych sygnału, a to jest istotą filtracji sygnału, jest też źródłem aliasingu. Brak utraty efektywności obliczeniowej przy rezygnacji z decymacji, został osiągnięty w konstrukcji zaproponowanej w artykule przez wykorzystanie rekursywnych struktur FIR. W odróżnieniu od klasycznych transwersalnych struktur filtrów, w tym przypadku liczba potrzebnych operacji jest niezależna od długości odpowiedzi impulsowych filtrów składowych. Stąd jest możliwe tworzenie banku złożonego z filtrów o różnych długościach odpowiedzi impulsowych i szerokościach pasm bez wpływu na liczbę wymaganych operacji arytmetycznych.

Konstrukcję podstawowej, komplementarnej pary filtrów dolnoprzepustowego i górnoprzepustowego oparto bezpośrednio na algorytmie rekursywnej transformaty Fouriera. W celu poprawy ich parametrów częstotliwościowych zastosowano okno trójkątne, które jest realizowane rekursywnie. W artykule przedstawiony został również sposób rekonstrukcji sygnału wyjściowego.

Zaproponowane banki filtrów posiadają liniową charakterystykę fazową i właściwość perfekcyjnej rekonstrukcji.

Słowa kluczowe: bank filtrów, rekursywna transformata Fouriera, rekursywna struktura FIR, aliasing

KWARTALNIK ELEKTRONIKI I TELEKOMUNIKACJI, 2006, 52, z. 1, ss. 49–72

Dithering methods in A/D conversion for uniform quantizers and errors introduced by dither

TOMASZ ADAMSKI

Institute of Electronic Systems
Warsaw Univesity of Technology
00-665 Warszawa ul. Nowowiejska 15/19
e-mail: T.Adamski@ise.pw.edu.pl

Otrzymano 2005.05.06
Autoryzowano 2005.07.19

In the paper mathematical background of the classical dithering method (used in A/D converters) is described. Some generalizations of the classical method are proposed. The new concept is based on inverting "dithering characteristic" F (defined in the paper). Both classical and generalized method errors are assessed and analyzed in detail.

Keywords: dithering, A/D conversion, quantizers, measurement accuracy, stochastic processes

1. INTRODUCTION

It is well known that dither (i.e. noise purposely added to the converted input voltage) can improve accuracy of A/D converters. In the paper we try to assess errors and accuracy of the classical dithering method and we propose a generalized dithering method. The essence of the new approach consist in averaging and inverting a "dithering characteristic" (defined in the sequel).

In general dithering methods are divided into two categories:

a) subractive dither (when we subtract dither realization after quantization),

b) non-subtractive dither (when we do not subtract dither realization after quantization).

The paper deals only with the non-subtractive dither.

Every constant inside intervals, real function of a real variable $QN : R \to R$ is called a quantization function. More precisely, we assume in the definition that we have a sequence $(I_i)_{i \in Z}$ of intervals $I_i \subset R$ that for every $i \in Z$ and $x \in I_i$ we have

$QN(x) = const.$ and $\bigcup_{i \in Z} I_i = R$ and there is a real number $\varepsilon > 0$ that for every $i \in Z$ we have $l_1(I_i) > \varepsilon$, where l_1 is a Lebesgue measure on R. Additionally we assume that for every $i, j \in Z$, $i \neq j$ we have $I_i \cap I_j = \emptyset$ and there exist two points $x_1, x_2 \in R$ that $QN(x_1) \neq QN(x_2)$.

From electronic engineering point of view the quantization function is a static input/output characteristic of an electronic circuit (with one or more comparators) called a quantizer.

The quantization function $QN : R \to R$ assumed in the sequel describes a typical A/D ideal rounding converter and is defined with the formula $QN(x) = \Delta x \left\lfloor \frac{1}{\Delta x} x + 1/2 \right\rfloor$ where $\Delta x > 0$ is a parameter. It is so called "the uniform quantization function" (see. Fig. 1.1).

Let D be a real random variable describing dither. Assume D is defined on a probabilistic space $(\Omega, \mathfrak{M}, P)$ and $QN(a + D) \in L^1(\Omega, \mathfrak{M}, P)$ for etery $a \in R$. In the paper a function $F : R \to R$ given for every $a \in R$ with the formula

$$F(a) \stackrel{df}{=} E(QN(a + D) = \int_R QN(a + x) P_D(dx) \tag{1}$$

is called a "dithering characteristic". P_D is a probability distribution of the random variable D. Then the dithering characteristic depends on two parameters: quantization function $QN : R \to R$ and probability distribution P_D of the random variable D.

In the section 2 we assess properties of this function. In particular we prove that under natural assumptions the function F is continuous and strictly monotonic increasing. Some particular examples of dithering characteristics are given in the section 3.

The section 4 deals with classical and generalized dithering method. The section 4 is devoted to errors and accuracy of the dithering methods.

A simplified block diagram of the circuit for measurement (i.e. A/D conversion) with the classical dithering method is shown in the Fig. 2. To the measured input voltage a is added a dither D and then the sum a + D is quantized and averaged.

Assume we have a random variable $D \in L^1(\Omega, \mathfrak{M}, P)$ or more strictly a sequence $(D_n)_{n=1}^{\infty}$ of independent random variables with the same distribution as the distribution of the random variable D (we assume, that $D = D_1$). In this case we have of course $D_n \in L^1(\Omega, \mathfrak{M}, P)$ for every $n \in N$. Note that $QN(a + D)$ is a random variable because a quantization function QN is measurable as a constant (inside intervals) function. Similarly for every $n \in N$, $QN(a + D_n)$ is a random variable. Additionally (proof in the section 2) we have $QN(a + D) \in L^1(\Omega, \mathfrak{M}, P)$ and $QN(a + D_n) \in L^1(\Omega, \mathfrak{M}, P)$ for every $n \in N$.

In N_0 experiments we obtain a sequence of numbers

$$QN(a + D_1)(\omega), QN(a + D_2)(\omega), ..., QN(a + D_{No})(\omega) \tag{2}$$

where $\omega \in \Omega$ is an elementary event. Then we have N_0 independent realizations of the random variable $QN(a+D)$ or N_0 first coordinates of the trajectory of the stochastic process $(QN(a+D_n))_{n=1}^{\infty}$. Denote $A(N_0) \stackrel{df}{=} \frac{1}{N_0}\sum_{n=1}^{N_0} QN(a+D_n)$. Now we compute the value

$$A(N_0)(\omega) = \frac{1}{N_0}\sum_{n=1}^{N_0} QN(a+D_n)(\omega). \tag{3}$$

Using Strong Law of Large Numbers (SLLN) (see appendix) we obtain that if $N_0 \to \infty$ then

$$\frac{1}{N_0}\sum_{n=1}^{N_0} QN(a+D_n)(\omega) \to E(QN(a+D) \quad P \text{ almost everywhere} \tag{4}$$

Under above mentioned (and some additional) assumptions on the random variable D distribution and the random sequence $(D_n)_{n=1}^{\infty}$ (see sections 2, 3 and 4) we have $E(QN(a+D) = a$ and we can admit $A(N_0)(\omega)$ as a value of the measured input voltage a. It is the essence of the dithering method.

The crucial question about the measured value a is answered in the following way: the value admitted as a (i.e. final result of measurement) is a realization of the random variable $A(N_0)$. It is exactly like in the case of parameter estimation in mathematical statistics, $A(N_0)(\omega)$ is an estimate of the value a. Speed of convergence of the estimator $A(N_0)$ to the measured value a is analyzed in the section 5.

In the paper we assess errors of the dithering method caused by

1) limited number of averaged samples N_0

2) the fact that a distribution of the random variable D can be different from assumed one.

In the sequel we describe also a generalization of the classical dithering method. The essence of the proposed solution consist in construction of a strong consistent and unbiased estimator of the value a for a larger class of dither D distributions. In the classical dithering method the class of acceptable distributions is limited to special kinds of uniform distributions. In generalized dithering method the class of acceptable distributions is (as it is proven below) much wider. The generalized method allows to estimate a with averaging and inverting the dithering characteristic. To prove correctness of the method we use Strong Law of Large Numbers and some properties of dithering characteristics.

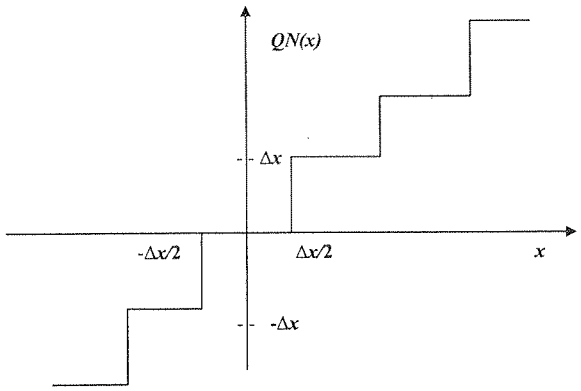


Fig. 1. Quantization function $QN : R \to R$ assumed in the paper is the mid-tread quantizer transfer characteristic

Rys. 1. Funkcja kwantyzacji $QN : R \to R$ rozważana w pracy jest charakterystyką typu "mid-tread" (czyli charakterystyką z przedprożem w środku)

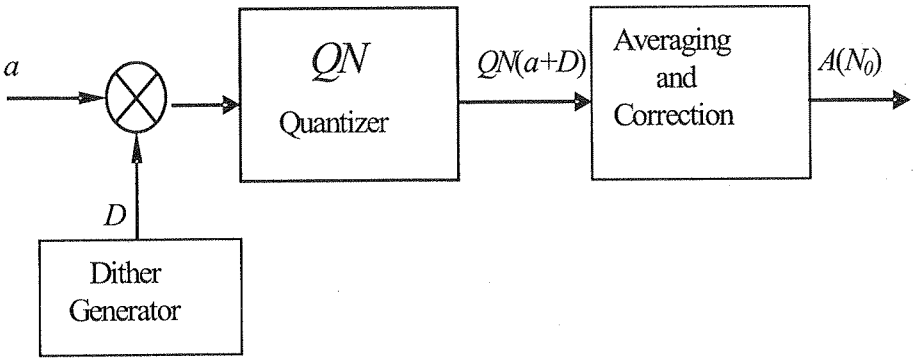


Fig. 2. Simplified block diagram of the circuit for measurement (i.e. A/D conversion) with classical and generalized dithering method. The dither D is added to the measured input voltage a and then the sum $a + D$ is quantized. The procedure is repeated N_0 times then quantized samples are averaged. The last step is correction of the average with the dithering characteristic, the output $A(N_0)$ is an estimator of the value a

Rys. 2. Uproszczony schemat blokowy układu do konwersji A/C wykorzystującej metodę ditheringu (klasyczną lub uogólnioną). Dither D jest dodawany do mierzonego napięcia wejściowego a. Następnie suma $a + D$ jest kwantowana. Procedura ta powtarzana jest N_0 razy a następnie skwantowane próbki są uśredniane. Ostatnim krokiem jest korekcja uzyskanej wartości średniej za pomocą charakterystyki ditheringu. Wartość wyjściowa $A(N_0)$ jest estymatorem wartości mierzonej a

2. BASIC PROPERTIES OF DITHERING CHARACTERISTICS

In the section we give some basic properties of the dithering characteristic defined in the section 1. We would like to know if the dithering characteristic is monotonic increasing, strictly monotonic increasing or continuous.

Theorem 2.1. Assume $QN : R \to R$ is a uniform quantization function. If a random variable D has the mean value i.e. $D \in L^1(\Omega, \mathfrak{M}, P)$ then
1) for every $a \in R$ there is the mean value $E(QN(a + D))$ (then the dithering characteristic is well defined)
2) a dithering characteristic is always a monotonic increasing function
(it is independent from the distribution of the random variable D describing dither, the distribution can be for example discrete, continuous or have an arbitrary type of the distribution)

Proof. Ad 1. We have to prove that the function $QN(a + D) : \Omega \to R$ (where a is an arbitrary real number) is P integrable. At first observe that $QN(a + D) : \Omega \to R$ is a random variable. A uniform quantization function is measurable as real function constant inside intervals. The function $a + D$ is of course measurable then a superposition $QN(a + D) : \Omega \to R$ is a measurable function too.

Introduce 2 functions $F_1 : R \to R$ and $F_2 : R \to R$ defined with the following formulas: $F_1(x) = -\frac{\Delta x}{2} + x$ and $F_2(x) = \frac{\Delta x}{2} + x$. They are of course measurable functions and for every $x \in R$ we have $F_1(x) \leqslant QN(x) \leqslant F_2(x)$. Then $F_1(a + D)$ and $F_2(a + D)$ are random variables and

$$F_1(a + D) \leqslant QN(a + D) \leqslant F_2(a + D) \tag{5}$$

It is easy to note that if $D \in L^1(\Omega, \mathfrak{M}, P)$ then also $F_1(a + D), F_2(a + D) \in L^1(\Omega, \mathfrak{M}, P)$. Indeed $F_1(a + D) \in L^1(\Omega, \mathfrak{M}, P)$ because

$$E(F_1(a + D)) = E\left(-\frac{\Delta x}{2} + a + D\right) = E\left(-\frac{\Delta x}{2} + a\right) + E(D) \tag{6}$$

In similar way we can prove, that $F_2(a + D) \in L^1(\Omega, \mathfrak{M}, P)$. Then $QN(a + D) \in L^1(\Omega, \mathfrak{M}, P)$ as a random variable bounded by a P integrable function (see (5)).

Ad.2. Indeed if $a_1 \leqslant a_2$ then of course $QN(a_1 + D) \leqslant QN(a_2 + D)$. Therefore $E(QN(a_1 + D)) \leqslant E(QN(a_2 + D)$ i.e. $F(a_1) \leqslant F(a_2)$ then a function F is monotonic increasing.■

Theorem 2.2. Let $QN : R \to R$ be a uniform quantization function. If a random variable D has the mean value i.e. $D \in L^1(\Omega, \mathfrak{M}, P)$ and has a density function then
1) for every $a \in R$ there is a mean value $E(QN(a + D)$
2) the dithering function $R \ni a \to F(a) = E(QN(a + D) \in R$ is continuous and monotonic increasing.

Proof. Partially the theorem is a consequence (or rather repetition) of the theorem 5. We have to prove only that the dithering characteristic is continuous. To prove that the

dithering characteristic $R \ni a \to F(a) = E(QN(a + D) \in R$ is continuous we have to prove that: if $a_n \to a_0$ then

$$E(QN(a_n + D) \to E(QN(a_0 + D) \tag{7}$$

or equivalently that

$$\lim_{n \to +\infty} E(QN(a_n + D) - QN(a_0 + D)) = 0 \tag{8}$$

but because there is a density of the random variable D then we have

$$E(QN(a_n + D) - QN(a_0 + D)) = \int_R (QN(a_n + x) - QN(a_0 + x))f(x)l_1(dx) \tag{9}$$

Note that if $a_n \to a_0$ then l_1 almost everywhere (l_1 is a Lebesgue measure on the real axis) we have

$$\lim_{n \to +\infty} (QN(a_n + x) - QN(a_0 + x)) = 0 \tag{10}$$

because we subtract "mutually close" shifted quantization functions. Then of course we have also: l_1 almost everywhere

$$\lim_{n \to +\infty} ((QN(a_n + x) - QN(a_0 + x))f(x)) = 0 \tag{11}$$

Note now that the absolute value of the integrated functions on the right side of the formula (9) can be bounded with a l_1 integrable function. Indeed, we have

$$|(QN(a_n + x) - QN(a_0 + x))f(x)| \leqslant (|QN(a_n + x)| + |QN(a_0 + x)|)f(x) \leqslant$$

$$2 \cdot |QN(m + x)|\, f(x)$$

where $m = \sup_{n \in N} a_n$.

From the theorem 5 it follows that $QN(m + D) \in L^1(\Omega, \mathfrak{M}, P)$ and of course $2 \cdot QN(m + D) \in L^1(\Omega, \mathfrak{M}, P)$. Then from the theorem on changing variables we have $2 \cdot QN(m + id) \cdot f \in L^1(R, \mathfrak{L}, l_1)$, where $id : R \to R$ is an identity function. As a result we have that an absolute value of the integrated functions in the formula (9) is bounded for every $n \in N$ by an integrable function. Then assumptions of the Lebesgue theorem (on convergence under the integral) are fulfilled. Therefore we have

$$\lim_{n\to+\infty} E(QN(a_n+D)-QN(a_0+D)) = \lim_{n\to+\infty}\int_R (QN(a_n+x)-QN(a_0+x))f(x)l_1(dx) =$$

$$= \int_R \lim_{n\to+\infty}(QN(a_n+x)-QN(a_0+x))f(x)l_1(dx) = 0$$

It proves the convergence (8) then the dithering characteristic F is continuous.■

Comment. In the above proof we use in fact the assumption that the random variable D has the density f related to the Lebesgue measure.

Theorem 2.3. Assume $QN : R \to R$ is a uniform quantization function. If a random variable D has the mean value i.e. $D \in L^1(\Omega, \mathfrak{M}, P)$ and has the density function $f \in L^1(R, \mathfrak{L}, l_1)$ such that for every $x \in [-\frac{1}{2}\Delta x, \frac{1}{2}\Delta x]$ (Δx is a parameter of the quantization function) we have $f(x) > 0$ then

1) for every $a \in R$ there is a mean value $E(QN(a+D)$
2) dithering characteristic $R \ni a \to F(a) = E(QN(a+D) \in R$ is continuous and strictly monotonic increasing

Proof. Partially the theorem 7 is a conclusion from the theorem 6. We have to prove only that the dithering characteristic $F : R \to R$ is a strictly monotonic function. Note that if $a_1 < a_2$ and $a_2 - a_1 < \Delta x/2$ then the function

$$R \ni x \to QN(a_2+x) - QN(a_1+x) \in R$$

is "a periodic sequence of rectangular impulses" with the width $a_1 - a_2$ and height $\geqslant \Delta x$. Additionally one impulse is placed completely on the interval $[-\frac{1}{2}\Delta x, \frac{1}{2}\Delta x]$.

From the Luzin theorem (from measure and integral theory) and the Stone-Weierstrass theorem (on properties of continuous functions on a compact set) it follows that the integral of the "sole impulse" is greater than 0 i.e.

$$\int_{[-\frac{1}{2}\Delta x, \frac{1}{2}\Delta x]} (QN(a_2+x)-QN(a_1+x))f(x)l_1(dx) > 0.$$

Hence we have

$$F(a_2)-F(a_1) = E(QN(a_2+D)-QN(a_1+D)) = \int_R (QN(a_2+x)-$$

$$-QN(a_1+x))f(x)l_1(dx) \geqslant \int_{[-\frac{1}{2}\Delta x, \frac{1}{2}\Delta x]} (QN(a_2+x)-QN(a_1+x))f(x)l_1(dx) > 0$$

It proves that the dithering characteristic function F is strictly monotonic.■

Theorem 2.4 Assume $p \geqslant 1$, $p \in R^+$, $a \in R$ and $QN : R \to R$ is a uniform quantization function. If $D \in L^p(\Omega, \mathfrak{M}, P)$ then $QN(a + D) \in L^p(\Omega, \mathfrak{M}, P)$.

Proof. Let c be such a constant that $c > \Delta x/2$. Consider a function $g : R \ni x \to g(x) \stackrel{df}{=} c + |x| \in R$. It is easy to verify that the following inequality holds.: $|QN(x)| \leqslant g(x)$, then also $|QN(x + a)| \leqslant g(x + a)$ and $|QN(x + D)| \leqslant g(x + D)$.

To prove, that $QN(a + D) \in L^p(\Omega, \mathfrak{M}, P)$ it is sufficient to show that $g(a + D) \in L^p(\Omega, \mathfrak{M}, P)$.

From the theory of B spaces $L^p(\Omega, \mathfrak{M}, P)$ it follows, that a constant function belongs to this space and if $D \in L^p(\Omega, \mathfrak{M}, P)$ then $a + D \in L^p(\Omega, \mathfrak{M}, P)$ and $|a + D| \in L^p(\Omega, \mathfrak{M}, P)$. Directly from this fact we obtain that $g(a + D) \in L^p(\Omega, \mathfrak{M}, P)$ and finally $QN(a + D) \in L^p(\Omega, \mathfrak{M}, P)$.■

3. SOME PARTICULAR EXAMPLES OF DITHERING CHARACTERISTICS

There are some easy to analyze particular cases of random variable D probability distributions.

Example 1. If dither is described with the Dirac measure concentrated in the point 0 then the dithering characteristic F is equal to a quantization function i.e. $F(x) = QN(x)$ for every $x \in R$. As a result the dithering characteristic is a monotonic increasing function but is not continuous.

If dither is described with the Dirac measure concentrated in the point $b \in R$ then the dithering characteristic F is given for every $x \in R$ by the formula $F(x) = QN(x + b)$. As a result the dithering characteristic is a monotonic increasing function but is not continuous.■

Example 2. If a random variable D has a discrete distribution concentrated on a finite number of points then the dithering characteristic F is a monotonic function, constant inside intervals but not continuous.■

Example 3. If a random variable D has a density f such that supp f is a subset of the closed interval $[-k_1\Delta x + b, k_2\Delta x + b]$, where $k_1, k_2 \in R^+$, $b \in R$ and $0 < k_1, k_2 < \frac{1}{2}$ then the dithering characteristic is continuous but constant on a number of intervals. Thus it is not invertible (see Fig. 6 and 8).■

Example 4. If a random variable D has a uniform distribution then dithering characteristic is continuous, monotonic increasing and linear inside intervals or linear.■

Example 5. If D has a uniform distribution on the interval $\left[-\frac{1}{2}k\Delta x + b, \frac{1}{2}k\Delta x + b\right]$ for fixed $k \in N$ and $b \in R$ then the dithering characteristic is given for every $x \in R$ by the formula $F(x) = x + b$ (see Fig. 7).

In particular if D has a uniform distribution on the interval $\left[-\frac{1}{2}k\Delta x, \frac{1}{2}k\Delta x\right]$ for fixed $k \in N$ then the dithering characteristic is linear i.e. $F(x) = x$ for every $x \in R$ (see Fig. 6, and 7).■

Example 6. If D has a probability distribution f given by a formula

$$f(x) = \sum_{k=-\infty}^{+\infty} \alpha_k \chi_{[-\frac{1}{2}k\Delta x, \frac{1}{2}k\Delta x]}$$

(where $\alpha_k \in R^+$ and $\chi_{[-\frac{1}{2}k\Delta x, \frac{1}{2}k\Delta x]}$ is a characteristic function of the set $\left[-\frac{1}{2}k\Delta x, \frac{1}{2}k\Delta x\right]$) then the dithering characteristic is linear i.e. $F(x) = x$ for every $x \in R$. Then a uniform distribution on the interval $\left[-\frac{1}{2}k\Delta x, \frac{1}{2}k\Delta x\right]$ for fixed $k \in N$ is not a unique type of distribution giving a linear dithering characteristic.■

Example 7. If D has a probability distribution f given by a formula

$$f(x) = \sum_{k=-\infty}^{+\infty} \alpha_k \chi_{[-\frac{1}{2}k\Delta x + b_k, \frac{1}{2}k\Delta x + b_k]}$$

where $\alpha_k \in R^+$, and $\chi_{[-\frac{1}{2}k\Delta x + b_k, \frac{1}{2}k\Delta x + b_k]}$ is a characteristic function of the set $\left[-\frac{1}{2}k\Delta x + b_k, \frac{1}{2}k\Delta x + b_k\right]$ then there exists $b \in R$ that the dithering characteristic is given for every $x \in R$ by the formula $F(x) = x + b$.■

Comment. For the given quantization function and probability distribution of the random variable D we can compute values of the dithering characteristic $F : R \to R$ with arbitrary accuracy. It is easy to note that $QN(x) = x - ([x + \Delta x/2]_{\Delta x} - \Delta x/2)$ then the quantization function is a sum of a linear function $id : R \to R$ and a periodic function $R \ni R \to -([x + \Delta x/2]_{\Delta x} - \Delta x/2) \in R$ with a period Δx. Therefore the value of the dithering characteristic $F(a)$ is equal to

$$F(a) = E(QN(a + D) = E(a + D) - E([a + D + \Delta x/2]_{\Delta x} + \Delta x/2 =$$
$$= a + E(D) - E([a + D + \Delta x/2]_{\Delta x} + \Delta x/2$$

then the function F is also composed from the linear part and the periodic part with a period Δx. Then for computation of the dithering characteristic F, it is sufficient to know only the following function

$$[-\Delta x/2, \Delta x/2] \ni a \to E([a + D + \Delta x/2]_{\Delta x})$$

In practice linearity (or nonlinearity) of the dithering characteristic is important. Nonlinearity of the dithering characteristic F (under assumption of the continuous differentiability F on the real axis R) can be defined as a number $\sup_{x \in R} F'(x) - \inf_{x \in R} F'(x)$.

It can be easily seen that:

$$\sup_{x\in R} F'(x) - \inf_{x\in R} F'(x) = \sup_{x\in[-\Delta x/2,\Delta x/2]} F'(x) - \inf_{x\in[-\Delta x/2,\Delta x/2]} F'(x) =$$

$$= \sup_{x\in[-\Delta x/2,\Delta x/2]} g'(x) - \inf_{x\in[-\Delta x/2,\Delta x/2]} g'(x)$$

where g is a periodic component of the dithering characteritic.

It follows from our computer simulations the following, intuitively clear conclusion: "larger dither" gives a "more linear" dithering characteristic F. This rule does not work always but is useful from practical point of view.

On Fig. 3–9 some exemplary dithering characteristics are shown.

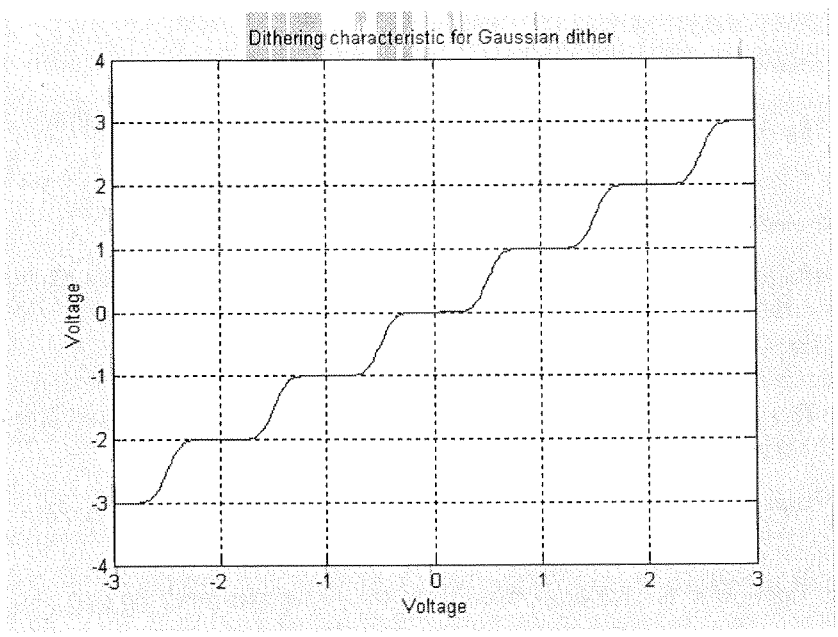


Fig. 3. Exemplary dithering characteristic when the dither distribution is Gaussian with the mean value 0 and a standard deviation $\sigma = 0.1\Delta x$. Quantization function is uniform with $\Delta x = 1$ V

Rys. 3. Przykład charakterystyki ditheringu dla gaussowskiego rozkładu ditheru z wartością średnią 0 i odchyleniem standardowym $\sigma = 0.1\Delta x$. Funkcja kwantyzacji jest równomierna z $\Delta x = 1$ V

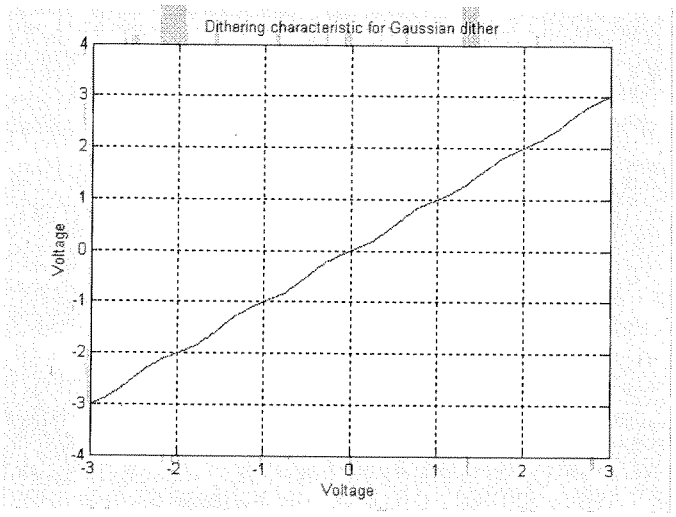


Fig. 4. Exemplary dithering characteristic when the dither distribution is Gaussian with the mean value 0 and a standard deviation $\sigma = 0.3\Delta x$. Quantization function is uniform with $\Delta x = 1$ V

Rys. 4. Przykład charakterystyki ditheringu dla gaussowskiego rozkładu ditheru z wartością średnią 0 i odchyleniem standardowym $\sigma = 0.3\Delta x$. Funkcja kwantyzacji jest równomierna z $\Delta x = 1$ V

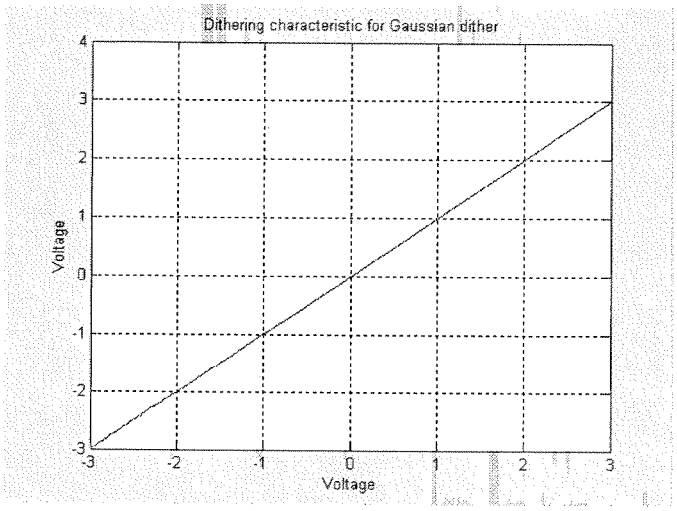


Fig. 5. Exemplary dithering characteristic when the dither distribution is Gaussian with the mean value 0 and a standard deviation $\sigma = 0.5\Delta x$. Quantization function is uniform with $\Delta x = 1$ V

Rys. 5. Przykład charakterystyki ditheringu dla gaussowskiego rozkładu ditheru z wartością średnią 0 i odchyleniem standardowym $\sigma = 0.5\Delta x$. Funkcja kwantyzacji jest równomierna z $\Delta x = 1$ V

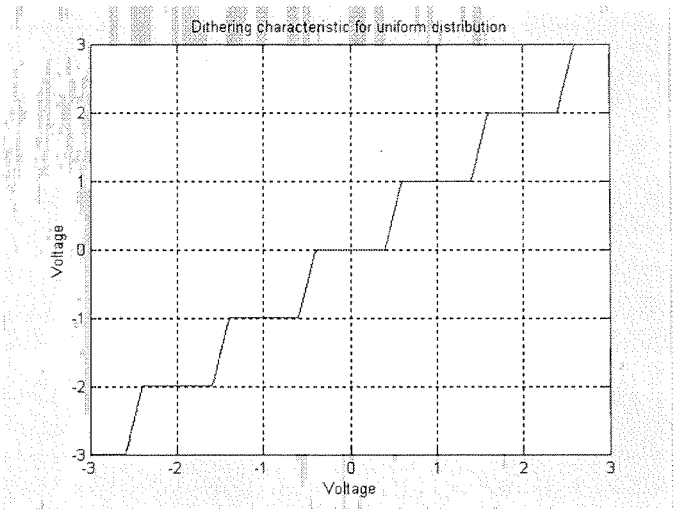


Fig. 6. Exemplary dithering characteristic when the dither distribution is uniform on the interval $[-0.1\ \Delta x,\ 0.1\ \Delta x]$. Quantization function is uniform with $\Delta x = 1$ V

Rys. 6. Przykład charakterystyki ditheringu dla jednostajnego rozkładu ditheru na przedziale $[-0.1\ \Delta x,\ 0.1\ \Delta x]$. Funkcja kwantyzacji jest równomierna z $\Delta x = 1$ V

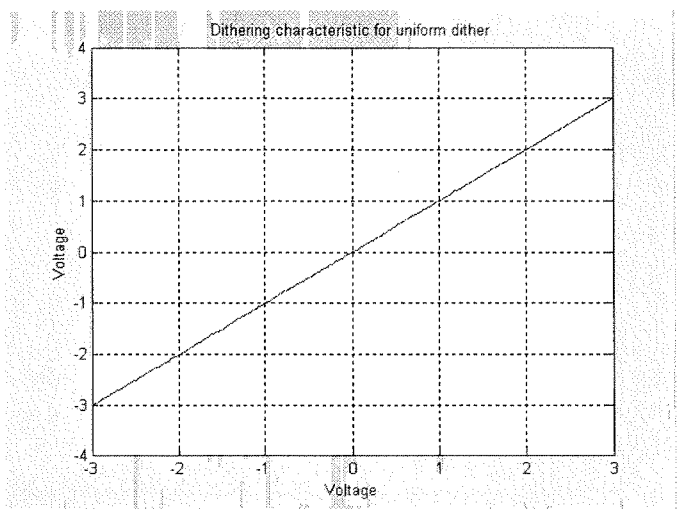


Fig. 7. Exemplary dithering characteristic when the dither distribution is uniform on the interval $\left[-\frac{1}{2}k\Delta x, \frac{1}{2}k\Delta x\right]$ for $k = 5.5$. Quantization function is uniform with $\Delta x = 1$ V

Rys. 7. Przykład charakterystyki ditheringu dla jednostajnego rozkładu ditheru na przedziale. $\left[-\frac{1}{2}k\Delta x, \frac{1}{2}k\Delta x\right]$ dla $k = 5.5$. Funkcja kwantyzacji jest równomierna z $\Delta x = 1$ V

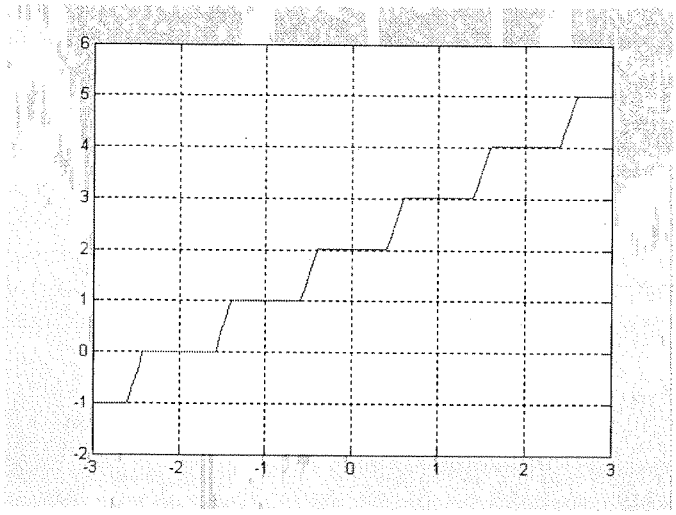


Fig. 8. Exemplary dithering characteristic when the dither distribution is uniform on the interval $[-0.1\Delta x + 2\Delta x, 0.1\Delta x + 2\Delta x]$. Quantization function is uniform with $\Delta x = 1$ V

Rys. 8. Przykład charakterystyki ditheringu dla jednostajnego rozkładu ditheru na przedziale $[-0.1\Delta x + 2\Delta x, 0.1\Delta x + 2\Delta x]$. Funkcja kwantyzacji jest równomierna z $\Delta x = 1$ V

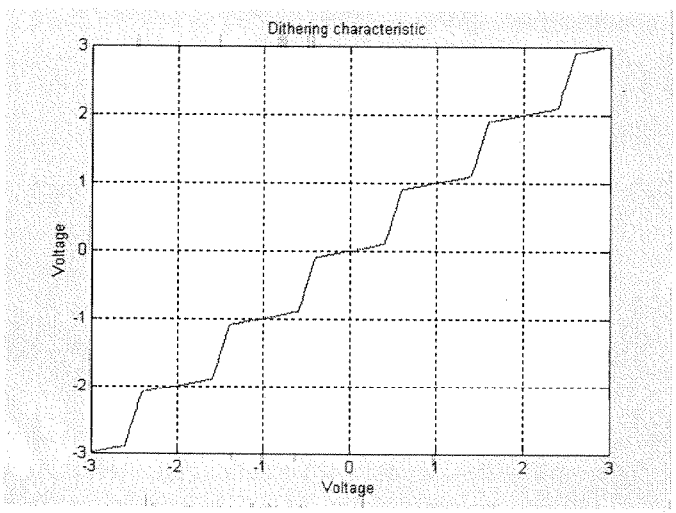


Fig. 9. Exemplary dithering characteristic when the dither distribution is given by the formula $f(x) = \frac{15}{4\Delta x}\chi_{[-0.1\Delta x, 0.1\Delta x]} + \frac{1}{4\Delta x}\chi_{[-0.5\Delta x, 0.5\Delta x]}$. Quantization function is uniform with $\Delta x = 1$ V

Rys. 9. Przykład charakterystyki ditheringu w sytuacji, gdy rozkład ditheru zadany jest wzorem $f(x) = \frac{15}{4\Delta x}\chi_{[-0.1\Delta x, 0.1\Delta x]} + \frac{1}{4\Delta x}\chi_{[-0.5\Delta x, 0.5\Delta x]}$ Funkcja kwantyzacji jest równomierna z $\Delta x = 1$ V

4. CLASSICAL AND GENERALIZED DITHERING METHODS

In the paper we assume that a random sequence $(D_n)_{n=1}^{\infty}$ (introduced in the section 1) is a sequence of independent random variables with the same probability distribution as the random variable D (we assume that $D = D_1$ and $D \in L^1(\Omega, \mathfrak{M}, P)$). In this case we have of course $D_n \in L^1(\Omega, \mathfrak{M}, P)$ for every $n \in N$. Hence for every $n \in N$ and every $a \in R$ we obtain $QN(a + D) \in L^1(\Omega, \mathfrak{M}, P)$ and $QN(a + D_n) \in L^1(\Omega, \mathfrak{M}, P)$.

Additionally assume, that the dithering characteristic $F : R \to R$ (see section 2) is continuous and strictly monotonic increasing. We can compute values of F with arbitrary accuracy computing values of the periodic component of the function F. To be exact we have to compute values only on the interval which has the length equal to the period of the periodic component.

The generalized dithering method is simple. We know the dithering characteristic $F : R \to R$ for a given distribution of the random variable D and we take N_0 values

$$QN(a + D_1)(\omega), QN(a + D_2)(\omega), ..., QN(a + D_n)(\omega), ..., QN(a + D_{N_0})(\omega) \tag{12}$$

where $\omega \in \Omega$ is an elementary event. Then we compute the mean value $\frac{1}{N_0} \sum_{n=1}^{N_0} QN(a + D_n)(\omega)$. A finite sequence (12) is composed of N_0 independent realizations of the random variable $QN(a + D)$ or in other words N_0 first coefficients of the trajectory of the stochastic process $(QN(a + D_n))_{n=1}^{\infty}$.

Using Strong Law of Large Numbers (SLLN) (see appendix) we obtain that P almost everywhere if $N_0 \to \infty$ then

$$\frac{1}{N_0} \sum_{n=1}^{N_0} QN(a + D_n)(\omega) \to E(QN(a + D) \tag{13}$$

i.e. P almost everywhere if $N_0 \to \infty$ then we have

$$\frac{1}{N_0} \sum_{n=1}^{N_0} QN(a + D_n)(\omega) \to F(a) \tag{14}$$

Because the function $F : R \to R$ (as strictly monotonic increasing and continuous) is invertible and an inverse function $F^{-1} : R \to R$ is continuous then from (14) we have P almost everywhere: if $N_0 \to \infty$ then

$$F^{-1}\left(\frac{1}{N_0} \sum_{n=1}^{N_0} QN(a + D_n)(\omega)\right) \to F^{-1}(F(a)) = a \tag{15}$$

As an estimator of the value a we can admit

$$A(N_0) \stackrel{df}{=} F^{-1}\left(\frac{1}{N_0}\sum_{n=1}^{N_0} QN(a + D_n)\right) \tag{16}$$

An estimate of the value a is then computed as a realization of the random variable $A(N_0)$ i.e.

$$A(N_0)(\omega) \stackrel{df}{=} F^{-1}\left(\frac{1}{N_0}\sum_{n=1}^{N_0} QN(a + D_n(\omega))\right) \tag{17}$$

The formula (15) says that the estimator (16) is strong consistent . From the convergence P almost everywhere it follows the asymptotic convergence of random variables (i.e. convergence in probability). In our case it means that if $N_0 \to +\infty$ then the convergence P almost everywhere $A(N_0) \to a$ implies the convergence $A(N_0) \to a$ in probability. Then we can say: for every $\varepsilon > 0$ and every $\delta > 0$ there is such $\tilde{N}_0 \in N$ that for every $N_0 \geqslant \tilde{N}_0$ we have

$$P(|A(N_0) - a| \geqslant \varepsilon) \leqslant \delta \tag{18}$$

In short, the answer about the value a is the following: a (a final result of the measurement) is in fact a realization of the random variable $A(N_0)$. $A(N_0)(\omega)$ is an estimate of the value a. It is exactly the same case as parameter estimation in mathematical statistics. The inequality (18) gives a good intuitive description of the "practical value" of the estimate $A(N_0)(\omega)$.

We can obtain the classical dithering method as a special case of the described above generalized dithering method.

If the random variable D describing dither has the uniform distribution on the interval $[-\frac{1}{2}k\Delta x, \frac{1}{2}k\Delta x]$ for fixed $k \in N$ then the dithering characteristic is linear i.e. $F(x) = x$ because under these assumptions $D \in L^1(\Omega, \mathfrak{M}, P)$ and we have

$$E(QN(a + D) = a \tag{19}$$

In this situation we have not to compute the inverse of the function F. The formula (19) is the essence of the classical dithering method. The equality (19) can be understood in the following way: averaging of samples cancels nonlinearity of the quantization function QN in the formula (19).

Finally in the classical dithering method we take as $A(N_0)(\omega)$ (an estimate of the value a) the average i.e.

$$A(N_0)(\omega) = \frac{1}{N_0}\sum_{n=1}^{N_0} QN(a + D_n(\omega)) \tag{20}$$

Comment. All over the paper we assume that $(D_n)_{n=1}^{\infty}$ is a sequence of independent random variables with the same probability distribution. The dithering method works

correctly also in the more general case when a sequence $(D_n)_{n=1}^{\infty}$ is a sequence of real random variables stationary in the strict sense. In both cases we assume that $(D_n)_{n=1}^{\infty}$ fulfills the additional condition: for every $n \in N$ we have $D_n \in L^1(\Omega, \mathfrak{M}, P)$ (see. Strong Law of Large Numbers in the appendix).

Asymmetry of the random variable D distribution causes errors in the classical dithering method. In generalized method it is not important because we correct these errors by inverting the dithering characteristic F. But in both dithering methods true information about a distribution P_D of the random variable D describing dither is a crucial point for accuracy.

When we do not use the true distribution of the random variable D (in dithering characteristic computations) it introduces usually a systematic error of the method.

Assume F_1 and F_2 are continuous and strictly monotonic increasing dithering characteristics. F_1 denotes a true dithering characteristic, and F_2 an admitted dithering characteristic.

For a systematic error of the method we can take a number $F_2^{-1}(F_1(a))-F_1^{-1}(F_1(a)) = F_2^{-1}(F_1(a)) - a$ but as a rule a is unknown then more convenient solution is taking as a systematic error (denote it by δ_0) an assessment done in the following way

$$\delta_0 \leqslant \sup_{x \in R} \left| F_2^{-1}(x) - F_1^{-1}(x) \right|.$$

We would like to prove the fact "if distributions of two random variables D^k and D describing dither are sufficiently "close" then dithering characteristic F_k and F (for D^k and D appropriately) are close too. The situation is explained by the following theorem".

Theorem 4.1 Assume a random variable $D \in L^1(\Omega, \mathfrak{M}, P)$ has a probability density and for every $k \in N$, $D^k \in L^1(\Omega, \mathfrak{M}, P)$ and a sequence $(D^k)_{k=1}^{\infty}$ is a uniform integrable family of random variables. If $D^k \to D$ converges weakly when $k \to +\infty$ then

1) for every $x \in R$, $F_k(x) \to F(x)$ when $k \to +\infty$ (point convergence), where F_k is a dithering characteristic for the random variable D^k and F is a dithering characteristic for the random variable D.

2) if for every $k \in N$ the random variable D^k has a density (related to the Lebesgue measure on the real axis) then $F_k \to F$ uniformly when $k \to +\infty$.

Comment 1. Assumption of weak convergence is not an especially restrictive assumption. If $D^k \to D$ P almost everywhere when $k \to +\infty$, or $D^k \to D$ in probability or $D^k \to D$ in the norm of $L^p(\Omega, \mathfrak{M}, P)$ (where $p \geqslant 1$) then $D^k \to D$ converges weakly when $k \to +\infty$.■

Comment 2. There are many natural examples of uniform integrability. For instance a sequence $(D^k)_{k=1}^{\infty}$ is a family of uniform integrable random variables if there is such a bounded interval $I \subseteq R$, that $P_{D^k}(I) = 1$ for every $k \in N$. Another example, if $D^k \in L^p(\Omega, \mathfrak{M}, P)$ where $p \geqslant 1$ and $D^k \to D$ converges in $L^p(\Omega, \mathfrak{M}, P)$ when $k \to +\infty$ to a random variable $D \in L^p(\Omega, \mathfrak{M}, P)$ then a family of random variables $(D^k)_{k=1}^{\infty}$ is uniform integrable.■

Proof. Ad. 1. We have to show, that for every $a \in R$ we have

$$\int_{\Omega} QN(a + D^k)dP \to \int_{\Omega} QN(a + D)dP$$

providing $k \to +\infty$. Equivalently: for every $a \in R$ when $k \to +\infty$ we have

$$\int_{R} QN(a + x)P_{D^k}(dx) \to \int_{R} QN(a + x)P_D(dx)$$

From the weak convergence $D^k \to D$ (i.e. from weak convergence of probability distributions: $P_{D^k} \to P_D$ when $k \to +\infty$) it follows (see [2]) that for every $A \in B(R)$ fulfilling the condition $P_D(\partial A) = 0$ we have:

$$\lim_{k \to +\infty} P_{D^k}(A) = P_D(A)$$

where ∂A denotes a border of the set A. Therefore for every interval $I_i = [i\Delta x - \Delta x/2, i\Delta x + \Delta x/2)$ for $i \in Z$ we have

$$\int_{I_i} QN(a + x)P_{D^k}(dx) \to \int_{I_i} QN(a + x)P_D(dx)$$

Denote $B_r = \bigcup_{i=-r}^{r} I_i$. From the above formula it follows that

$$\int_{B_r} QN(a + x)P_{D^k}(dx) \to \int_{B_r} QN(a + x)P_D(dx)$$

From the uniform integrability of the family of random variables $(D^k)_{k=1}^{\infty}$ we obtain the uniform integrability of the family of random variables $(QN(a+D^k))_{k=1}^{\infty}$. Therefore we have

$$\int_{R} QN(a + x)P_{D^k}(dx) \to \int_{R} QN(a + x)P_D(dx).$$

Hence $F_k(a) \to F(a)$ when $k \to +\infty$ which proves the point convergence i.e. the first part of the thesis.

Ad.2 Under admitted assumptions dithering characteristics are continuous. On the other hand (see comments on periodicity in the section 2) dithering characteristics F_k and F can be written as $F_k(x) = id(x) + g_k(x)$ and $F(x) = id(x) + g(x)$, where g_k and g are continuous functions with a period Δx. Thus $|F_k - F|$ is a periodic function with a period Δx and we have

$$\sup_{x\in R}|F_k(x)-F(x)| = \sup_{x\in[-\Delta x/2,\Delta x/2]}|F_k(x)-F(x)| = \sup_{x\in[-\Delta x/2,\Delta x/2]}|g_k(x)-g(x)|$$

Because a sequence of continuous functions which converges in every point on the closed interval is uniformly convergent then for an arbitrary $\varepsilon > 0$ there is such $N_0 \in N$, that for every $k \geqslant N_0$ we have $\sup_{x\in[-\Delta x/2,\Delta x/2]}|g_k(x)-g(x)| < \varepsilon$ and finally $\sup_{x\in R}|F_k(x)-F(x)| < \varepsilon$. It proves the uniform convergence $F_k \to F$ when $k \to +\infty$.■

5. SPEED OF CONVERGENCE OF THE ESTIMATOR $A(N_0)$ TO THE MEASURED VALUE a

Assume the assumptions of the section 4 (concerning the random variable D which describes dither) are fulfilled. Assume additionally that, $D \in L^2(\Omega, \mathfrak{M}, P)$ (equivalently we can say that there is a variance of the random variable D). From results of the section 2 we have $QN(a+D) \in L^2(\Omega, \mathfrak{M}, P)$ and there is a variance $D^2(QN(a+D))$.

As a value of the measured voltage a we admit a number:

$$A(N_0)(\omega) \stackrel{df}{=} F^{-1}\left(\frac{1}{N_0}\sum_{n=1}^{N_0} QN(a+D_n(\omega))\right) \tag{21}$$

Quality of the estimator $A(N_0)$ (i.e. its errors) can be assessed with the variance $D^2(A(N_0))$. As will be proved in the sequel the variance $D^2(A(N_0))$ can be assessed with a sequence which tends to 0 when $N_0 \to +\infty$. Therefore we can control the value of the variance (increasing number N_0) and taking N_0 suited to the needed accuracy.

As was proved in the section 4 it follows from Strong Law of Large Numbers that $A(N_0) \to a$ when $N_0 \to +\infty$ (P almost everywhere) and also $A(N_0) \to a$ (in probability) (which is in electronic measurement practice is more intuitive), see the inequality (18).

The inequality (18) does not tell us how to choose N_0 for an acceptable level of accuracy. Such a mechanism gives the Chebyshev inequality (see appendix). From the Chebyshev inequality we obtain that for an arbitrary $\varepsilon > 0$ we have

$$P(|A(N_0)-a| \geqslant \varepsilon) \leqslant \frac{D^2(A(N_0))}{\varepsilon^2} \tag{22}$$

If we prove the convergence $D^2(A(N_0)) \to 0$ when $N_0 \to +\infty$ then we will be able to control accuracy of estimation by choosing appropriate N_0.

We can easily assess the variance $D^2(A(N_0))$ assuming that the dithering characteristic F is continuously differentiable on R and $F'(x) > 0$ for every $x \in R$ (then F is strictly monotonic increasing). Denote $A_0 \stackrel{df}{=} \inf_{t\in R} F'(t)$. We have in this situation

$$A_0 \stackrel{df}{=} \inf_{t\in R} F'(t) = \inf_{t\in[-\Delta x/2,\Delta x/2]} F'(t) = 1 + \inf_{t\in[-\Delta x/2,\Delta x/2]} g'(t) > 0, \tag{23}$$

where g is a periodic component of the dithering characteristic. In the case of classical dithering method we have $A_0 = 1$.

$$D^2(A(N_0)) = D^2\left(F^{-1}\left(\frac{1}{N_0}\sum_{n=1}^{N_0} QN(a+D_n)\right)\right) \leqslant D^2\left(\frac{1}{A_0N_0}\sum_{n=1}^{N_0} QN(a+D_n)\right) = \tag{24}$$

$$= \frac{1}{A_0^2N_0^2}D^2\left(\sum_{n=1}^{N_0} QN(a+D_n)\right) = \frac{1}{A_0^2N_0}D^2(QN(a+D_n)$$

Thus the assessment of the variance $D^2(A(N_0))$ is inversely proportional to N_0 and $D^2(A(N_0)) \to 0$ when $N_0 \to +\infty$. The obtained result can be formulated in this way: "accuracy of the dithering method is proportional to $\sqrt{N_0}$".

6. RANDOMLY CHANGING COMPARISON LEVEL OF THE QUANTIZER

In real electronic circuits, comparison levels of quantizers are noised i.e are changing randomly. There are small random fluctuations around a mean value.

Assume, that random fluctuations of the quantization level are described by a random variable L with a continuous probability distribution and the random variables L and D are independent.

We can "add" these random fluctuations to the dither introducing a random variable $D' = D + L$ instead of the random variable D. A distribution of the "new" dither, i.e. a distribution of the random variable $D' = D + L$ is a convolution of two probability distributions i.e.

$$P_{D'} = P_D * P_L \tag{25}$$

If f_1 denotes a density of the random variable D and f_2 a density of the random variable L then a random variable $D' = D + L$ has a density f equal to the convolution $f = f_1 * f_2$ of two densities.

Random changes of quantizer comparison levels can be treated as dither and can be in natural way taken into account in the generalized dithering method.

Consider now the case when the dither D has a discrete distribution. It is the case when we use as a dither generator a generator of pseudorandom numbers (see Fig.10).

A convolution of the discrete distribution with a continuous one is a continuous distribution. More precisely, assume P_D is a discrete probability distribution on the real axis R concentrated in points $(a_i)_{i=1}^r$ then the convolution $P_{D'} = P_D * P_L$ has a density given with the following formula

$$f(x) = \sum_{i=1}^{r} f_2(x-a_i)P(a_i) \tag{26}$$

where f_2 is a density of the random variable L. Similarly if P_D has a discrete probability distribution on the real axis R concentrated in points $(a_i)_{i=1}^{\infty}$ then the convolution $P_{D'} = P_D * P_L$ has a density given by the following formula

$$f(x) = \sum_{i=1}^{\infty} f_2(x-a_i)P(a_i) \tag{27}$$

As we see, random fluctuations of the comparison level can change a discrete distribution of the dither into a continuous one.

Typical circuit for dither generation is shown in the Fig. 10 As a generator of pseudorandom numbers we can use for example a BBS (Blum, Blum, Shrub) generator or an appropriately chosen LFSR (Linear Feedback Shift Register).

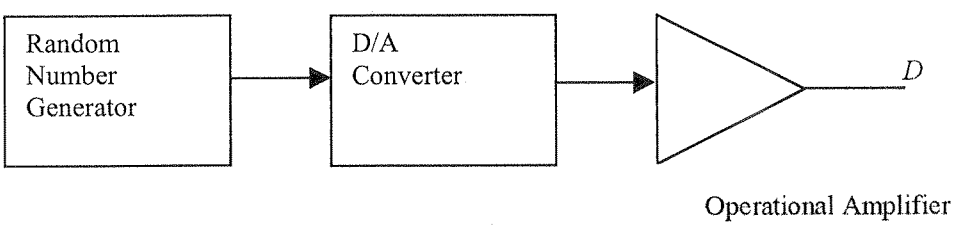


Fig. 10. A simple circuit for dither generation

Rys. 10. Prosty układ generatora ditheru

7. CONCLUSIONS

1. It is possible to generalize the classical dithering method on the case of much more wider family of dither distributions when compared with the classical dithering method.
2. A dithering characteristic F is very useful notion. It allows assessment of the accuracy of the classical and generalized dithering methods in particular we can easily assess systematic errors of the methods.
3. Limitation of the dithering method are accuracy of the generated distribution of the random variable D (real distribution of the random variable D can differ from admitted one) and non accurate identification of the random variable L describing fluctuations of the comparison level of the quantizer.
4. Under natural assumptions on the dither D, the variance $D^2(A(N_0))$ of the estimator $A(N_0)$ (of the measured voltage a) is inversely proportional to N_0, where N_0 is a number of averaged samples.

5. From many computer simulations it follows an intuitively clear conclusion: "greater dither" gives "more linear" dithering characteristic F. This rule does not work in general but is useful from practical point of view.
6. Random changes of comparison levels of the quantizer can be treated as dither and can be in natural way taken into account in the generalized dithering method.
7. In the paper we analyze a typical quantizer circuit. Similar results can be obtained for wider class of quantizers (see [5]) for example a two level quantizer with saturation.

8. REFERENCES

1. R. van de Plassche: *Integrated Analog-to-Digital and Digital-to-Analog Converters*, Kluwer Academic Publishers, 1994.
2. J. Jakubowski, R. Sztencel: *Introduction to Probability Theory* (in Polish), Script 2004.
3. Z. Kulka: *On Some Important Aspects of Digital Processing of Analog Signals in High Energy Physics Measurement Systems*, D.Sc. Thesis, Świerk, 1995.
4. K. Hejn: *Selected Metrological Problems of Analog to Digital Converters* (in Polish), Oficyna Wydawnicza P.W.; Warszawa 1999.
5. T. Adamski: *Generalized Dithering Method for A/D Converters*, Proceedings of the Polish - Hungarian-Czech Workshop on Circuit Theory, Signal Processing and Applications; Prague September 2003.
6. T. Owczarek: *How to Choose Accuracy of A/D Converters when Dithering is Applied* (in Polish), KKE, Kołobrzeg 2004.

List of symbols

ω – elementary event
N – set of natural numbers
R – set of real numbers
R^+ – set of real nonnegative numbers
$(\Omega, \mathfrak{M}, P)$ – probability space
$L^1(\Omega, \mathfrak{M}, P)$ – space of integrable real functions (random variables)
$L^p(\Omega, \mathfrak{M}, P)$ – space of integrable (with the power p, $p \geqslant 1$) real functions
$L^1(R, \mathfrak{L}, l_1)$ – space of l_1 integrable (on the real axis) real functions
l_1 – Lebesgue measure on the real axis
D – real random variable describing dither
$(D_n)_{n=1}^{\infty}$ – a sequence of real random variable describing dither
QN – quantization function
$E(X)$ – mean value of the random variable X
$D^2(X)$ – variance of a random variable X
a – a constant value of the converted voltage, $a \in R$
$A(N_0)$ – random variable, an estimator of the value a obtained from N_0 samples
$\lfloor x \rfloor$ – floor function value for the argument x
$id : R \to R$ – identity function
$[\ \cdot\ ]_b$ – a function modulo a nonnegative number b

$[\ x\ ]_b = \min\{y \in R^+\ ;$ it exists $k \in Z$ that $y = x - k \cdot b\}$
$P_3 = P_1 * P_2$ – convolution of two probability measures P_1 and P_2
supp f – suport of the function f i.e. supp $f = \overline{\{x \in X;\ f(x) = 0\}}$

Appendices

1. Strong Law of Large Numbers

Theorem (strong law of large numbers for stationary processes in strict sense) If $(X(n))_{n\in Z}$ is a real stochastic process stationary in strict sense and defined on a probabilistic space $(\Omega, \mathfrak{M}, P)$ and $X(n) \in L^1(\Omega, \mathfrak{M}, P)$ for every $n \in Z$, then P almost everywhere

$$\left(\sum_{n=1}^{N_0} X(n)\right)/N_0 \to E(X(0)) \qquad \text{when} \quad N_0 \to +\infty$$

and P almost everywhere

$$\left(\sum_{n=-N_0}^{N_0} X(n)\right)/(2N_0 + 1) \to E(X(0)) \qquad \text{when} \quad N_0 \to +\infty$$

Comment 1. As a corollary from the above theorem we obtain a strong law of large numbers in classical formulation i.e. for a sequence of independent random variables $(X(n))_{n\in N}$ with the same distribution, having a variance (i.e. square integrable). In this case the following equality also holds

$$D^2((X(1) + X(2) + \ldots + X(N_0))/N_0) = \frac{1}{N_0} D^2(X(1)).$$

Comment 2. Theorem D.2 is also true for stationary processes in strict sense with values in Banach spaces.

2. Markov and Chebyshev inequality

Markov inequality. If $X \in L^p(\Omega, \mathfrak{M}, P)$ where $p \geqslant 1$ then for every $\varepsilon > 0$ we have

$$P(|X| \geqslant \varepsilon) \leqslant \frac{E(|X|^p)}{\varepsilon^p}$$

Directly from the Markov inequality we obtain the Chebyshev inequality.
Chebyshev inequality. If a random variable $X \in L^2(\Omega, \mathfrak{M}, P)$ then for every $\varepsilon > 0$ we have

$$P(|X - E(X)| \geqslant \varepsilon) \leqslant \frac{D^2(X)}{\varepsilon^2}$$

T. ADAMSKI

METODY DITHERINGU W KONWERSJI A/D DLA KWANTYZERÓW RÓWNOMIERNYCH I BŁĘDY WPROWADZANE PRZEZ DITHER

Streszczenie

Jak wiadomo dither czyli szum o odpowiednich parametrach celowo dodawany do konwertowanego na wartość cyfrową napięcia umożliwia podwyższenie dokładności przetworników A/D.

W pracy została zaproponowana uogólniona metoda ditheringu. Uogólniona metoda ditheringu sprowadza się do:
– wstępnego obliczenia dla danego kwantyzera zdefiniowanej w artykule charakterystyki ditheringu F oraz funkcji odwrotnej F^{-1} do charakterystyki ditheringu
– uśrednienia ciągu skwantowanych próbek z dodanym ditherem a następnie
– obliczenia wartości funkcji F^{-1} dla otrzymanej średniej

Oznaczmy przez $QN : R \to R$ funkcję kwantyzacji równomiernej typu mid-tread tzn. $QN(x) = \Delta x \left\lfloor \frac{1}{\Delta x} x + 1/2 \right\rfloor$. Niech ponadto D oznacza zmienną losową rzeczywistą (opisującą dither) określoną na pewnej przestrzeni probabilistycznej $(\Omega, \mathfrak{M}, P)$ a P_D rozkład prawdopodobieństwa zmiennej losowej D. Funkcję $F : R \to R$ zdefiniowaną dla każdego $a \in R$ wzorem

$$F(a) \stackrel{df}{=} E(QN(a + D)) = \int_R QN(a + x) P_D(dx)$$

będziemy nazywać charakterystyką ditheringu. Zatem charakterystyka ditheringu zależy od 2 parametrów: funkcji kwantyzacji $QN : R \to R$ i rozkładu prawdopodobieństwa P_D zmiennej losowej D.

W rozdziale 2 pracy badane są szczegółowo własności charakterystyki ditheringu. Pokażemy, że przy naturalnych założeniach funkcja F jest ciągła i monotoniczna a więc odwracalna. Rozdział 3 poświęcony jest podstawom matematycznym klasycznej i uogólnionej metodzie ditheringu.

W pracy zostały również ocenione źródła i wielkość błędów klasycznej i uogólnionej metody ditheringu.

Słowa kluczowe: dithering, konwersja A/D, kwantyzery, dokładość pomiaru, procesy stochastyczne

KWARTALNIK ELEKTRONIKI I TELEKOMUNIKACJI, 2006, 52, z. 1, ss. 73–87

Szeregowanie niezależnych, wywłaszczalnych zadań periodycznych jedno- i dwuprocesowych z wykorzystaniem metody super zadań

MIROSŁAW GAJER

Katedra Automatyki, Akademia Górniczo-Hutnicza
al. Mickiewicza 30, 30-059 Kraków
e-mail: mgajer@ia.agh.edu.pl

Otrzymano 2004.10.18
Autoryzowano 2005.01.19

Artykuł został poświęcony problematyce szeregowania zbioru niezależnych i wywłaszczalnych zadań periodycznych. Zadania o takich właściwościach należą do najbardziej rozpowszechnionych we wszelkiego rodzaju systemach czasu rzeczywistego o ostrych ograniczeniach czasowych. W literaturze przedmiotu zaproponowano wiele różnych algorytmów służących do generacji planów szeregowania rozważanych zadań, które są w stanie zagwarantować dotrzymanie ograniczenia czasowego przez każde z zadań podlegających szeregowaniu. Jednym z najpowszechniej stosowanych algorytmów szeregujących jest algorytm RMS (ang. Rate Monotonic Scheduling), który bazuje na priorytetach przypisywanych zadaniom w ten sposób, że zadania o krótszych wartościach okresu otrzymują wyższy priorytet. Pewnym ograniczeniem algorytmu RMS jest to, iż w swej klasycznej postaci nadaje się wyłącznie do szeregowania zadań przeznaczonych tylko dla jednego procesora. Tymczasem obecnie coraz większą popularność zaczynają zdobywać rozwiązania wieloprocesorowe. W związku z powyższym w artykule została przedstawiona propozycja rozszerzenia obszaru stosowalności algorytmu RMS, tak aby mógł posłużyć również do szeregowania zadań przeznaczonych dla dwóch procesorów. Autor zaproponował wykorzystanie algorytmu genetycznego w celu optymalizacji rozdziału zadań do poszczególnych procesorów, tak aby zapewnić jak najmniejsze obciążenie jednostki przetwarzającej dane, co zwiększa szansę na szeregowalność zbioru zadań.

Słowa kluczowe: szeregowanie zadań, zadania wieloprocesorowe, algorytm szeregujący RMS

1. WPROWADZENIE

Systemy czasu rzeczywistego stanowią obecnie już dobrze wyodrębnioną klasę systemów komputerowych przewidzianych głównie do specjalistycznych zastosowań przemysłowych i telekomunikacyjnych [1]. Najważniejszym czynnikiem odróżniającym systemy czasu rzeczywistego o tzw. ostrych ograniczeniach czasowych (ang. hard real-time) [6] od systemów komputerowych ogólnego przeznaczenia jest nałożenie na realizację każdego z zadań ściśle określonego ograniczenia czasowego, którego przekroczenie jest niedopuszczalne, ponieważ może prowadzić do utraty kontroli nad sterowanym obiektem, co może poskutkować katastrofą, wielkimi stratami finansowymi, a nawet utratą życia ludzi [2]. Takie postawienie sprawy wymusiło silny rozwój formalnych metod projektowania systemów czasu rzeczywistego, których podstawowym celem jest zagwarantowanie, że podczas pracy systemu nigdy nie dojdzie do naruszenia ograniczenia czasowego przez żadne z występujących w nim zadań [3], [4].

W systemach czasu rzeczywistego o ostrych ograniczeniach czasowych szeregowaniu podlega najczęściej zbiór zadań periodycznych [5]. Ponadto szeregowane zadania są zwykle wzajemnie niezależne, tzn. mogą być realizowane w dowolnej kolejności, ponieważ wyniki żadnego z zadań nie stanowią danych wejściowych dla zadań pozostałych, oraz posiadają cechę wywłaszczalności, czyli wykonywanie każdego z zadań może zostać przerwane w dowolnym momencie jego realizacji, a następnie zostać wznowione po odtworzeniu ze stosu procesora kontekstu wywłaszczonego zadania.

Jednym z najpowszechniej stosowanych algorytmów do szeregowania zbioru niezależnych i wywłaszczalnych zadań periodycznych jest algorytm RMS (ang. Rate Monotonic Scheduling), który w przypadku spełnienia przez szeregowany zbiór zadań pewnych warunków jest w stanie zagwarantować dochowanie ograniczenia czasowego przez każde z zadań.

2. ALGORYTM SZEREGUJĄCY RMS

Istota działania algorytmu szeregującego RMS opiera się na kilku prostych zasadach. Przede wszystkim zadaniom przydzielane są priorytety, w ten sposób, że im krótszy okres posiada dane zadanie, tym wyższy jest jego priorytet [5]. Spośród zadań będących w stanie gotowości do realizacji zawsze wykonywane jest zadanie o najwyższym priorytecie. Jeżeli podczas wykonywania dowolnego zadania, w stan gotowości wejdzie jakieś inne zadanie o wyższym priorytecie, wówczas wykonywane aktualnie zadanie zostanie wywłaszczone, a procesor zostanie przekazany zadaniu o wyższym priorytecie [9]. Wywłaszczone zadanie może zostać wznowione jedynie w przypadku, gdy już żadne inne zadanie o wyższym od niego priorytecie nie znajduje się w stanie gotowości [6].

Każde zadanie szeregowane algorytmem RMS charakteryzowane jest przez swój czas wykonywania C oraz okres T. Iloraz C/T określa stopień wykorzystania czasu pracy procesora przez rozważane zadanie. Ponieważ szeregowaniu podlega zbiór N

niezależnych i wywłaszczalnych zadań periodycznych, suma $\sum_{i=1}^{N} \frac{C_i}{T_i}$ określa całkowity stopień wykorzystania czasu pracy procesora przez szeregowany zbiór zadań.

W 1973 roku Liu i Layland udowodnili ważne twierdzenie, zgodnie z którym, jeżeli spełniona jest nierówność $\sum_{i=1}^{N} \frac{C_i}{T_i} \leqslant N\left(2^{\frac{1}{N}} - 1\right)$, wówczas zbiór zadań jest szeregowalny, tzn. każde zadanie zawsze dochowa swego ograniczenia czasowego [15]. Rozważane twierdzenie stanowi warunek wystarczający na szeregowalność zbioru zadań, nie jest ono natomiast warunkiem koniecznym, w związku z czym jego nie spełnienie jeszcze niczego nie przesądza. W takim wypadku należy sprawdzić czas zakończenia realizacji każdego z zadań dla najgorszego przypadku (tzn. w sytuacji, gdy wszystkie zadania wchodzą jednocześnie w stan gotowości), i jeżeli każde z zadań zakończy swoją realizację przed upływem ograniczenia czasowego, to oznacza, iż rozważany zbiór zadań jest szeregowalny [7].

Czas zakończenia realizacji zadania, w sytuacji gdy równocześnie z nim zostało aktywowanych N-1 zadań o wyższych priorytetach oblicza się za pomocą następującej procedury iteracyjnej. Jako pierwsze przybliżenie przyjmuje się sumę czasów realizacji wszystkich zadań, ponieważ zanim może zostać rozpoczęta realizacja zadania o najniższym priorytecie, każde z zadań o priorytetach wyższych musi zostać wykonane co najmniej jeden raz. Zatem $t_0 = \sum_{i=1}^{N} C_i$. Kolejne przybliżenia czasu zakończenia zadania o najniższym priorytecie wyznaczane są jako kolejne iteracje $t_{k+1} = \sum_{i=1}^{N} C_i \left\lceil \frac{t_k}{T_i} \right\rceil$, do momentu, w którym spełniony zostanie warunek $t_{k+1} = t_k$.

Przedstawione obliczenia należy wykonać dla każdego z zadań, poczynając od zadania o najniższym priorytecie, a na zadaniu o najwyższym priorytecie kończąc. Dopiero dochowanie ograniczeń czasowych przez wszystkie zadania uprawnia do wyciągnięcia wniosku odnośnie szeregowalności zbioru zadań [8].

3. ROZSZERZENIE OBSZARU STOSOWALNOŚCI ALGORYTMU RMS DLA ZADAŃ WIELOPROCESOROWYCH

W swej klasycznej postaci algorytm RMS przeznaczony jest tylko i wyłącznie do szeregowania zadań jednoprocesorowych. Natomiast artykuł niniejszy stanowi propozycję rozszerzenia obszaru stosowalności algorytmu RMS dla systemów zbudowanych z dwóch jednostek obliczeniowych. Systemy takie stają się obecnie coraz bardziej popularne, bowiem coraz częściej można spotkać płyty główne z dwoma procesorami [10] oraz wszystko wskazuje na to, że już w najbliższej przyszłości powszechne zastosowanie znajdą procesory dwurdzeniowe, gdzie w postaci pojedynczego układu scalonego zintegrowane zostały dwie jednostki obliczeniowe [11].

Algorytmy szeregujące zadania wieloprocesorowe dzielą się na dwie klasy algorytmów przeznaczonych do szeregowania zadań przewidzianych dla procesorów dedykowanych oraz procesorów arbitralnych. W przypadku szeregowania zadań przeznaczonych dla procesorów dedykowanych jest z góry ustalone, na jakim procesorze dane zadanie ma być wykonywane. Ograniczenie to nie obowiązuje w przypadku procesorów arbitralnych, gdzie przydział zadania do procesora jest dowolny [12].

Zaproponowane rozszerzenie obszaru stosowalności algorytmu RMS dotyczy zarówno zadań przeznaczonych dla procesorów arbitralnych, jak i dedykowanych.

Istota propozycji autora polega na dokonaniu transformacji okresów wybranych zadań jednoprocesorowych, w taki sposób, aby kilka zadań jednoprocesorowych miało identyczną wartość okresu. W takim przypadku rozważane zadania mogą zostać połączone w jedno większe tzw. super zadanie, które pomimo tego, iż składa się z kilku mniejszych zadań jednoprocesorowych, przez system szeregujący może zostać potraktowane jako jedno zadanie dwuprocesorowe. Jeśli dokona się transformacji wszystkich zadań jednoprocesorowych w odpowiednie superzadania dwuprocesorowe, wówczas procesowi szeregowania będą podlegały tylko i wyłącznie zadania wieloprocesorowe przeznaczone dla dwóch procesorów. Aby możliwe było bezpośrednie zastosowanie algorytmu RMS do szeregowania zbioru niezależnych, wywłaszczalnych i periodycznych zadań dwuprocesorowych, należy je potraktować, jako zadania jednoprocesorowe, do szeregowania których potrzebna jest specjalna jednostka przetwarzająca zbudowana z dwóch procesorów. Jednakże sposób wewnętrznej budowy jednostki przetwarzającej jest nieistotny dla programu szeregującego, który wszystkie zadania traktuje jako zadania jednoprocesorowe, do których stosuje bezpośrednio algorytm RMS.

Zaproponowana metoda szeregowania zbioru niezależnych, wywłaszczalnych i periodycznych zadań jedno- i dwuprocesorowych została zilustrowana na przykładach, dotyczących przypadku zadań przeznaczonych dla procesorów dedykowanych i arbitralnych.

4. PRZYKŁAD SZEREGOWANIA ZADAŃ PRZEZNACZONYCH DLA JEDNEGO LUB DWÓCH PROCESORÓW

Niech dany będzie zbiór niezależnych, wywłaszczalnych i periodycznych zadań jedno- i dwuprocesorowych o charakterystykach zamieszczonych w tabeli 1. W kolejnych kolumnach tabeli 1 zamieszczono numer zadania, informację o typie zadania (1 – zadanie jednoprocesorowe, 2 – zadanie dwuprocesorowe), wartość okresu zadania oraz czas realizacji zadania.

W celu zastosowania algorytmu RMS do szeregowania zbioru zadań o charakterystykach zamieszczonych w tabeli 1 należy dokonać transformacji okresów zadań t_2 i t_3, w taki sposób, aby były one równe okresowi zadania t_1. Transformacja okresu zadania polega na zmniejszeniu jego wartości, dzięki czemu zadanie jest aktywowane nieco częściej. Postępowanie takie sprawia, że sterowany obiekt zachowuje się w sposób bardziej stabilny, ponieważ jego parametry będące przedmiotem sterowania korygowane

są częściej, a zatem odtwarza on zadaną trajektorię sterowania z większą dokładnością. Jedynym ujemnym wpływem zmniejszenia wartości okresu zadania jest wzrost stopnia wykorzystania czasu pracy procesora, a zatem zadanie po transformacji okresu będzie nieco w większym stopniu obciążało procesor niż przed transformacją.

Tabela 1

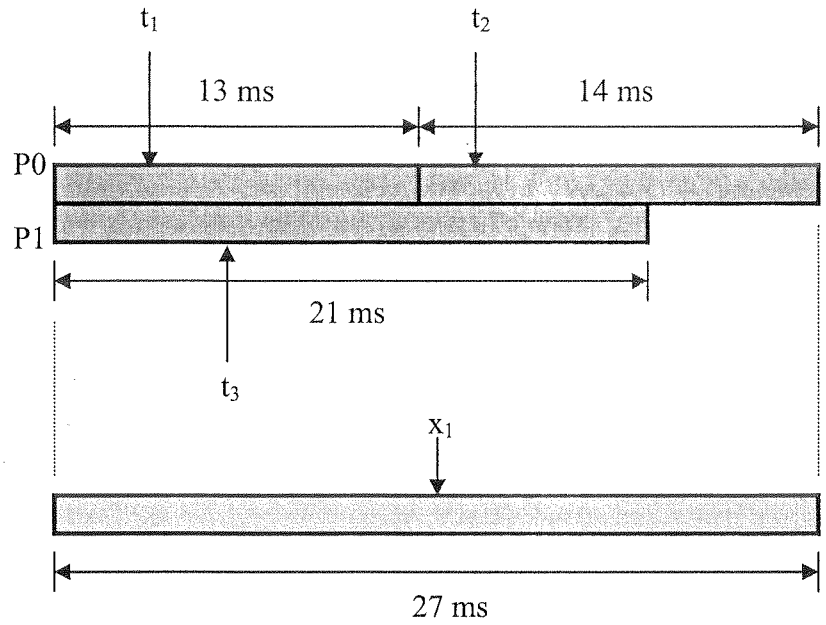
Zbiór niezależnych, wywłaszczalnych i periodycznych zadań jedno- i dwuprocesorowych

The set of independent, pre-emptive, and periodic uni- and biprocessor tasks

zadanie	typ zadania	okres zadania [ms]	czas realizacji [ms]
t_1	1	238	13
t_2	1	246	14
t_3	1	267	21
t_4	2	280	9
t_5	1	304	13
t_6	1	326	16
t_7	1	356	23
t_8	1	379	20
t_9	2	390	7
t_{10}	1	406	16
t_{11}	1	420	8
t_{12}	1	432	11
t_{13}	2	445	10
t_{14}	1	470	15
t_{15}	1	500	8
t_{16}	1	530	9
t_{17}	2	548	12
t_{18}	1	590	24
t_{19}	1	635	14
t_{20}	1	689	13

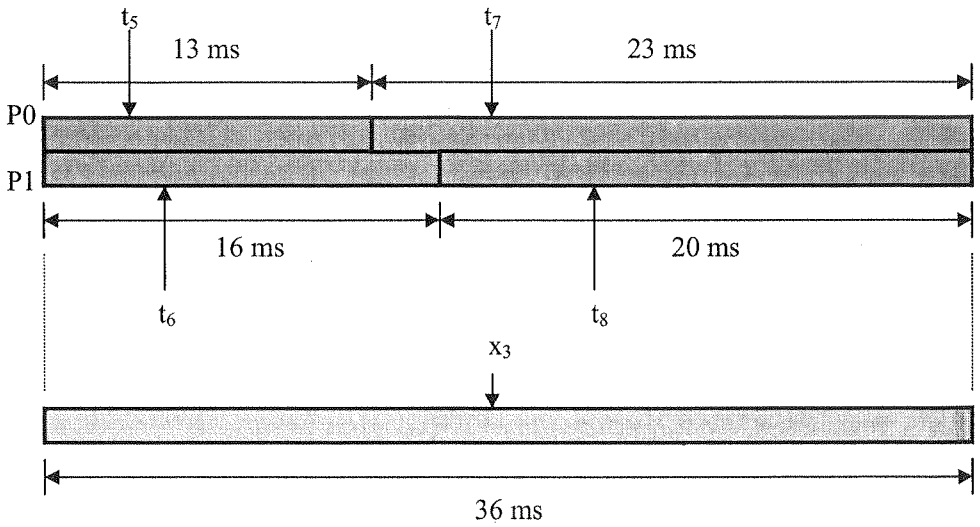
W wyniki transformacji okresów zadania t_1, t_2 i t_3 uzyskały taką samą wartość okresu równą 238 ms. Rozważane zadania zostały połączone w jedno super zadanie dwuprocesorowe x_1 w sposób przedstawiony na rys. 2.

Następnie zadanie dwuprocesorowe t_4 traktowane jest jako super zadanie x_2 o okresie 280 ms. Z kolei zadania t_5, t_6, t_7 i t_8 tworzą super zadanie x_3 w sposób przedstawiony na rys. 2.



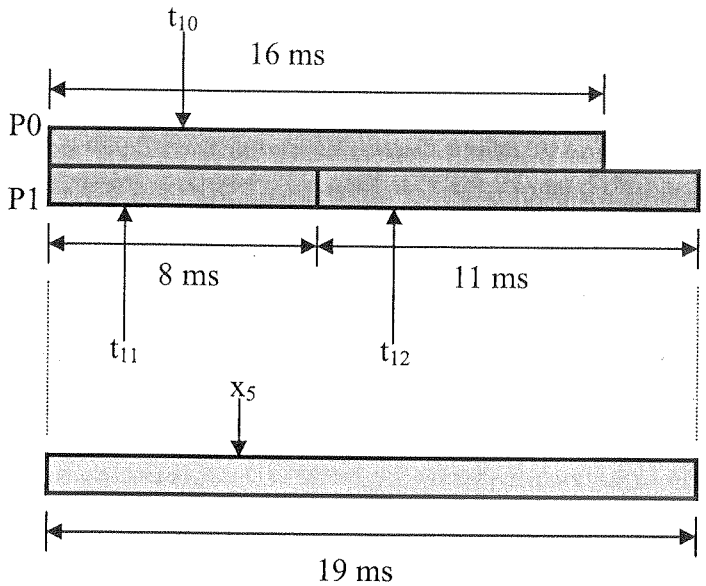
Rys. 1. Super zadanie x_1 utworzone z trzech zadań jednoprocesorowych t_1, t_2 i t_3

Fig. 1. Super task x_1 made of three uniprocessor tasks t_1, t_2 i t_3



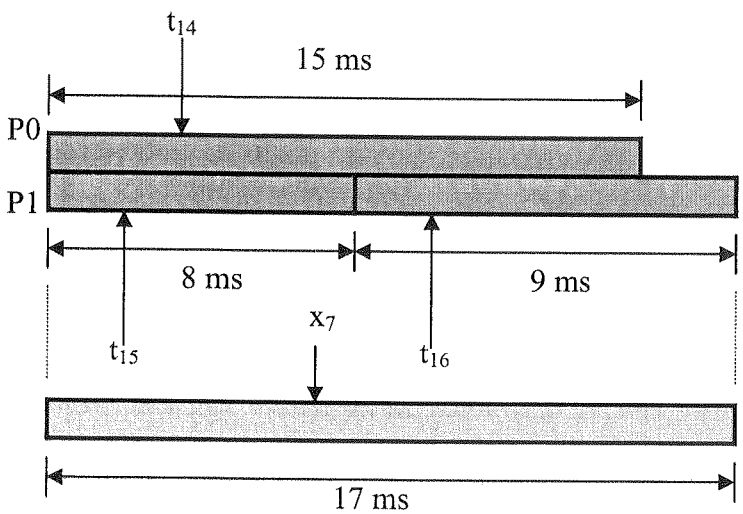
Rys. 2. Super zadanie x_3 utworzone z czterech zadań jednoprocesorowych t_5, t_6, t_7 i t_8

Fig. 2. Super task x_3 made of four uniprocessor tasks t_5, t_6, t_7 i t_8



Rys. 3. Super zadanie x_5 utworzone z trzech zadań jednoprocesorowych t_{10}, t_{11} i t_{12}

Fig. 3. Super task x_5 made of three uniprocessor tasks t_{10}, t_{11} i t_{12}



Rys. 4. Super zadanie x_7 utworzone z trzech zadań jednoprocesorowych t_{14}, t_{15} i t_{16}

Fig. 4. Super task x_7 made of three uniprocessor tasks t_{14}, t_{15} i t_{16}

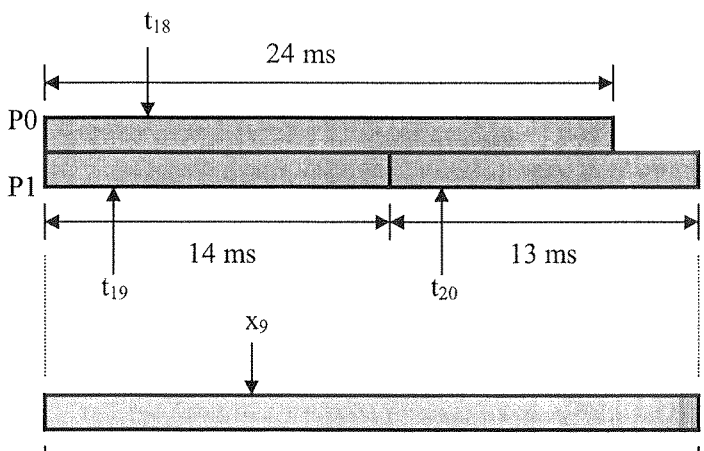
Super zadanie x_3 ma okres równy 304 ms, czyli tyle ile wynosi najmniejszy okres spośród zadań jednoprocesorowych t_5, t_6, t_7 i t_8 tworzących rozważane super zadanie.

Następne super zadanie x_4 zostaje utworzone z zadania dwuprocesorowego t_9 i ma okres równy 390 ms.

Z kolei super zadanie x_5 zostaje utworzone z połączenia zadań jednoprocesorowych t_{10}, t_{11} i t_{12} w sposób pokazany na rys. 3. Super zadanie x_5 ma okres 406 ms.

Super zadanie x_6 zostaje utworzone z zadania dwuprocesorowego t_{13} o okresie 445 ms. Z kolei super zadanie x_7 zostaje utworzone z trzech zadań jednoprocesorowych t_{14}, t_{15} i t_{16}. Okres super zadania x_7 wynosi 470 ms. Sposób połączenia zadań t_{14}, t_{15} i t_{16} został przedstawiony na rys. 4.

Z kolei super zadanie x_8 zostaje utworzone z zadania dwuprocesorowego t_{17} o okresie 548 ms. Natomiast ostatnie z super zadań x_9 zostaje utworzone z trzech zadań jednoprocesorowych t_{18}, t_{19} i t_{20} w sposób pokazany na rys. 5. Okres super zadania x_9 wynosi 590 ms.



Rys. 5. Super zadanie x_9 utworzone z trzech zadań jednoprocesorowych t_{18}, t_{19} i t_{20}

Fig. 5. Super task x_9 made of three uniprocessor tasks t_{18}, t_{19} i t_{20}

Charakterystyki wszystkich super zadań zostały zebrane w tabeli 2.

Całkowity stopień wykorzystania czasu pracy dwuprocesorowej jednostki przetwarzającej przez szeregowany zbiór super zadań wynosi 0,454. Jest to mniej niż prawa strona nierówności $\sum_{i=1}^{N} \frac{C_i}{T_i} \leqslant N\left(2^{\frac{1}{N}} - 1\right)$ wynosi dla N = 9, co daje wartość $9\left(2^{\frac{1}{9}} - 1\right) = 0,721$. Zatem na mocy odpowiedniego twierdzenia [8] rozważany zbiór super zadań jest szeregowalny na dwuprocesorowej jednostce przetwarzającej w przypadku zastosowania algorytmu RMS.

Tabela 2

Charakterystyki super zadań

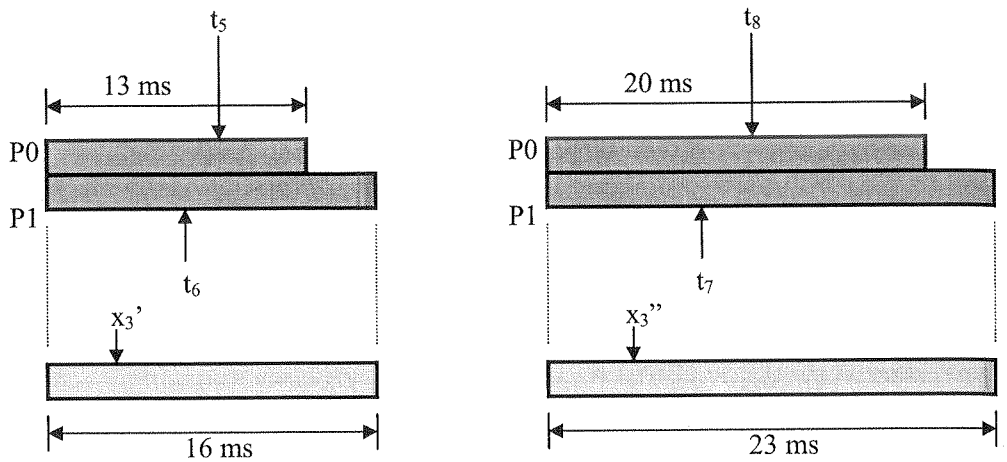
The characteristics of super tasks

super zadanie	okres super zadania [ms]	czas realizacji [ms]	stopień wykorzystania czasu pracy jednostki przetwarzającej
x_1	238	27	0,113
x_2	280	9	0,032
x_3	304	36	0,118
x_4	390	7	0,018
x_5	406	19	0,047
x_6	445	10	0,022
x_7	470	17	0,036
x_8	548	12	0,022
x_9	590	27	0,046

5. ZASTOSOWANIE ALGORYTMU GENETYCZNEGO

Jeżeli szeregowaniu podlega duża liczba zadań, wówczas istnieje znacznie większa swoboda odnośnie wyboru sposobu łączenia zadań jednoprocesorowych w dwuprocesorowe super zadania. Na przykład, w rozważanym w poprzednim punkcie przykładzie cztery zadania jednoprocesorowe t_5, t_6, t_7 i t_8 zostały połączone w jedno super zadanie x_3 o okresie 304 ms. Rozważane super zadanie wykorzystuje czas pracy dwuprocesorowej jednostki przetwarzającej w stopniu równym 0,118. Jednak warto zwrócić uwagę, że istnieje jeszcze jeden alternatywny sposób postępowania, w którym zadania t_5 i t_6 tworzą super zadanie x_3' o czasie realizacji 16 ms i okresie 304 ms, natomiast zadania t_7 i t_8 tworzą kolejne super zadanie x_3" o czasie realizacji 23 ms i okresie 356 ms. Ponieważ super zadanie x_3' wykorzystuje czas pracy jednostki przetwarzającej w stopniu 0,053, a super zadanie x_3" w stopniu równym 0,064, sumaryczne obciążenie dwuprocesorowej jednostki przetwarzającej przez oba super zadania x_3' i x_3" wynosi 0,117, co stanowi wynik o 0,001 lepszy niż uzyskano w przypadku utworzenia tylko jednego super zadania x_3, które łączyło w sobie wszystkie cztery zadania jednoprocesorowe. Sposób konstrukcji super zadań x_3' i x_3" pokazano na rys. 6.

W rozważanym przykładzie wybór metody tworzenia super zadań z zadań jednoprocesorowych t_5, t_6, t_7 i t_8 nie ma większego znaczenia, ponieważ sumaryczne obciążenie jednostki przetwarzającej przez wszystkie super zadania nie jest zbyt wielkie i w związku z tym istnieje znaczny margines bezpieczeństwa. Natomiast w pewnych przypadkach może się oczywiście tak zdarzyć, że jeden z alternatywnych sposobów



Rys. 6. Alternatywny sposób tworzenia super zadań

Fig. 6. The alternative way of super tasks construction

konstrukcji super zadań będzie prowadził do powstania zbioru szeregowalnego, inny natomiast nie.

W przypadku niewielkiej liczby zadań można zastosować metodę przeglądu zupełnego i sprawdzić wszystkie możliwe kombinacje połączeń zadań jednoprocesorowych w super zadania, a następnie wybrać spośród otrzymanych zbiorów super zadań te, które prowadzą do powstania zbioru szeregowalnego, a następnie wśród nich wyłonić ten, który obciąża jednostkę obliczeniową w najmniejszym stopniu.

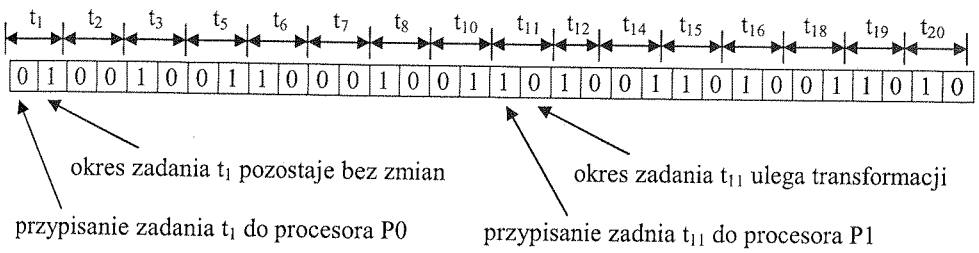
Sprawa znacznie się komplikuje, jeżeli szeregowaniu podlega duża liczba zadań jednoprocesorowych (np. sto i więcej). Wówczas metoda przeglądu zupełnego w ogóle, ze względu na swoją złożoność obliczeniową (kombinatoryczna eksplozja liczby możliwych rozwiązań), nie wchodzi w rachubę i trzeba, w związku z tym, zastosować heurystyczne metody przybliżone, które pozwolą na wyłonienie rozwiązania suboptymalnego o odpowiedniej jakości [14].

W celu znalezienia jak najlepszych sposobów połączeń poszczególnych zadań w super zadania, autor artykuł zdecydował się na wybór algorytmu genetycznego, jako rozwiązania łączącego w sobie zalety łatwości implementacji w postaci programu komputerowego, uniwersalności oraz dużej skuteczności w znajdowaniu rozwiązań o wymaganej jakości [12]. Podstawową zaletę algorytmu genetycznego stanowi również fakt, że w przypadku jego realizacji obliczenia mogą zastać przerwane praktycznie w dowolnym momencie i zawsze można z populacji osobników wyłonić najlepsze rozwiązanie. Zatem w przypadku algorytmu genetycznego jest możliwe dokonywanie wymiany pomiędzy jakością uzyskiwanych rozwiązań a czasem obliczeń poświęconym na ich otrzymanie.

Algorytm genetyczny działa na populacji wielu tysięcy osobników, na których wykonywane są cyklicznie operacje krzyżowania, mutacji i selekcji. W wyniku wykonywania wymienionych operacji genetycznych systematycznie wzrasta średnia wartość funkcji dopasowania liczona dla wszystkich osobników populacji. Wykonywanie obliczeń może zostać przerwane w momencie, gdy użytkownik stwierdzi, że uzyskiwane rozwiązania suboptymalne posiadają już zadowalającą jakość.

Każdy gen osobnika może przyjmować dwie wartości - zero lub jeden. W podejściu zaproponowanym przez autora liczba genów osobnika jest równa podwojonej liczbie zadań jednoprocesorowych występujących w szeregowanym zbiorze zadań. Zatem z każdym zadaniem jednoprocesorowym związane są dwa geny. Pierwszy z nich określa numer procesora, który realizował będzie dane zadanie. Jeśli rozważany gen ma wartość zero, wówczas zadanie zostanie przydzielone do procesora P0, w przypadku przeciwnym do procesora P1.

Z kolei drugi gen fragmentu materiału genetycznego związanego z danym zadaniem określa, czy wartość okresu tego zadania ulega transformacji, czy też pozostaje bez zmian. Jeśli rozważany gen ma wartość jeden, wówczas okres skojarzonego z nim zadania pozostaje bez zmian. W przypadku przeciwnym okres rozważanego zadania ulega zmniejszeniu do najbliższej wartości okresu zadania, który nie uległ zmianie. Na rys. 7 przedstawiono sposób zakodowania rozwiązania dla omówionego w poprzednim punkcie przykładu szeregowania zbioru zadań.



Rys. 7. Ilustracja sposobu kodowania rozwiązania na materiale genetycznym

Fig. 7. The ilustration of the way of coding of solution on the genetic material

Bardzo ważną sprawą przy opracowywaniu algorytmu genetycznego jest odpowiednie zdefiniowanie funkcji dopasowania, która służy do oceny jakości rozwiązań uzyskiwanych w wyniku cyklicznego stosowania operacji genetycznych. W podejściu zastosowanym przez autora wyznaczanie wartości funkcji dopasowania zostało podzielone na dwa odrębne przypadki.

Przypadek pierwszy dotyczy sytuacji, w której uzyskano zbiór zadań, który jest nieszeregowalny, tzn. istnieje w nim co najmniej jedno zadanie, które narusza swoje ograniczenie czasowe. W tym przypadku funkcji dopasowania zostaje przypisana war-

tość równa sumie czasów przekroczeń ograniczeń czasowych wszystkich zadań, które nie mogą zostać zakończone w terminie. Zostanie to zilustrowane na następującym przykładzie. Niech będzie dany zbiór pięciu zadań t_1, t_2, t_3, t_4, i t_5, z których trzy t_1, t_4 i t_5 naruszają swoje ograniczenia czasowe, odpowiednio t_1 o 12 ms, t_2 o 8 ms i t_1 o 15 ms. Wówczas funkcja dopasowania uzyskuje wartość f = 12 + 8 + 15 = 35. Oczywiście im mniejsza jest wartość funkcji dopasowania tym uzyskane rozwiązanie jest lepsze, ponieważ tym mniejsze jest sumaryczne naruszenie ograniczeń czasowych zadań. Zatem w procesie selekcji, jeżeli są porównywane dwa osobniki, dla których funkcja dopasowania została wyliczona zgodnie z procedurą odpowiednią dla przypadku pierwszego, powinien zwyciężyć osobnik o mniejszej wartości funkcji dopasowania.

Z kolei przypadek drugi dotyczy sytuacji, w której uzyskano szeregowalny zbiór zadań. Wówczas wartość funkcji dopasowania liczona jest w odmienny sposób, a mianowicie przypisana jej zostaje wartość sumarycznego stopnia obciążenia jednostki obliczeniowej (dwuprocesorowej) przez szeregowany zbiór zadań. Oczywiście im sumaryczne obciążenie jednostki obliczeniowej jest mniejsze, tym uzyskane rozwiązanie jest lepsze, ponieważ zadania mają większy margines zapasu czasu, jaki pozostał do upływu ich ograniczeń czasowych (ang. laxity), a ponadto jednostka obliczeniowa obciążona w mniejszym stopniu może zostać wykorzystana np. do realizacji pewnych zadań aperiodycznych, które zwykle występują w każdym systemie czasu rzeczywistego (stosowana jest wówczas technika zwana periodycznym serwerem zadań sporadycznych [13]). Zatem podczas selekcji, jeśli porównywane są ze sobą dwa osobniki, dla których wyznaczono wartość funkcji dopasowania zgodnie z procedurą odpowiednią dla przypadku drugiego, zawsze wygrywa osobnik o mniejszej wartości funkcji dopasowania.

Do omówienia pozostał jeszcze przypadek, w którym w procesie selekcji porównywane są ze sobą dwa osobniki, z których dla pierwszego wyznaczono wartość funkcji dopasowania zgodnie z procedurą opisaną w przypadku pierwszym (dotyczy to sytuacji, w której co najmniej jedno zadnie narusza swoje ograniczenie czasowe), natomiast dla drugiego wyznaczono wartość funkcji dopasowania zgodnie z procedurą opisaną w przypadku drugim (dotyczy to sytuacji, w której uzyskano szeregowalny zbiór zadań). W takim przypadku zawsze wygrywa osobnik drugi, ponieważ każde rozwiązanie, które prowadzi do szeregowalności zbioru zadań jest w oczywisty sposób zawsze lepsze od dowolnego innego rozwiązania, w którym choć jedno zadanie narusza swoje ograniczenie czasowe. Oczywiście funkcja dopasowania musi być dodatkowo wyposażona w zmienną logiczną, która będzie pozwalała na rozróżnienie dwóch różnych sposobów wyznaczania jej wartości (albo jako sumy czasów przekroczeń ograniczeń czasowych, albo jako sumarycznego obciążenia jednostki przetwarzającej).

6. ZAKOŃCZENIE

W artykule zamieszczono propozycję rozszerzenia obszaru stosowalności algorytmu RMS (ang. Rate Monotonic Scheduling) dla przypadku szeregowania zbioru niezależnych, wywłaszczalnych i periodycznych zadań przeznaczonych dla jednego i

dwóch procesorów. Rozszerzenie obszaru stosowalności algorytmu RMS dla rozważanego typu zadań jest ważne głównie ze względu na szybką ewolucję techniki wytwarzania mikroprocesorów, zmierzającą w kierunku układów dwurdzeniowych, gdzie w jednym układzie scalonym zintegrowane zostają dwie jednostki obliczeniowe. Układy takie znajdą z całą pewnością zastosowanie również w systemach czasu rzeczywistego o ostrych ograniczeniach czasowych, gdzie czas obliczeń jest najbardziej krytycznym z parametrów, a zapotrzebowanie na moc obliczeniową nieustannie rośnie, w związku ze stosowaniem coraz bardziej zaawansowanych algorytmów przetwarzania sygnałów pochodzących z sensorów, którymi opomiarowany jest obiekt sterowania.

Zaproponowane przez autora rozszerzenie obszaru stosowalności algorytmu RMS dla układów dwuprocesorowych zostało zilustrowane na przykładzie szeregowania zbioru zadań przeznaczonych dla arbitralnych procesorów. Jednak uzyskane wyniki łatwo mogą zostać przeniesione również na przypadek procesorów dedykowanych. W tym przypadku materiał genetyczny będzie posiadał o połowę mniejszą liczbę bitów, ponieważ nie będą występowały bity kodujące numer procesora, do którego dane zadanie ma zostać przydzielone (w przypadku procesorów dedykowanych przydział zadania do procesora jest określony z góry).

Istnieje również możliwość przeniesienia uzyskanych rezultatów na systemy zbudowane z większej liczby jednostek obliczeniowych (np. cztero-, sześcio- lub ośmioprocesorowe). W tym wypadku wystarczy zwiększyć jedynie liczbę bitów materiału genetycznego przeznaczonego do kodowania numeru jednostki obliczeniowej, do której zadanie jest przydzielane.

W przypadku zastosowania algorytmu genetycznego wyniki o zadawalającej jakości uzyskuje się dla populacji o liczebności tysiąca osobników i przy zastosowaniu krzyżowania dwupunktowego. Wykonywanie operacji genetycznych można przerwać po przeminięciu około pięciu tysięcy pokoleń, gdyż ewentualna dalsza poprawa jakości uzyskanego rozwiązania jest niewielka, a czas oczekiwania na polepszenie wyniku znacznie wzrasta. Autor wykonał eksperymenty szeregując zbiory drobnoziarnistych zadań o liczebności 100, 200 lub 500 zadań. Utworzone przez algorytm genetyczny superzadania wykorzystywały czas pracy jednostki obliczeniowej (dwuprocesorowej) w około 80%. Próba uszeregowania rozważanych super zadań w ponad 98% zakończyła się sukcesem.

7. BIBLIOGRAFIA

1. T. Szmuc: *Zaawansowane metody tworzenia oprogramowania systemów czasu rzeczywistego*, Krakowskie Centrum Informatyki Stosowanej, Kraków, 1998.
2. T. Szmuc: *Modele i metody inzynierii oprogramowania systemow czasu rzeczywistego*, AGH - Uczelniane Wydawnictwa Naukowo-Dydaktyczne, Kraków 2001.
3. T. Szmuc, G. Motet: *Specyfikacja i projektowanie oprogramowania czasu rzeczywistego*, Krakowskie Centrum Informatyki Stosowanej, Kraków, 1998.
4. T. Czachórski: *Modele kolejkowe w ocenie efektywności sieci i systemów komputerowych*, Wydawnictwo Pracowni Komputerowej Jacka Skalmierskiego, Gliwice, 1999.

5. A. S. Tanenbaum: *Rozproszone systemy operacyjne*, Wydawnictwo Naukowe PWN, Warszawa, 1997.
6. K. G. Shin, P. Ramanathan: *Real-time computing: A new discipline of computer science and engineering.* Proceedings of the IEEE, vol. 82, no. 1, January 1994, pp. 6-24.
7. K. Ramamritham, J. A. Stankovic: *Scheduling algorithms and operating systems support for real-time systems.* Proceedings of the IEEE, vol. 82, no. 1, January 1994, pp. 55-67.
8. A. Zalewski: *What every engineer needs to know about rate-monotonic scheduling: A tutorial.* Department of Computer Science, The University of Texas Of the Permian Basin, Odessa, TX, 1995, pp. 321-335.
9. A. Czajka, J. Nawrocki: *Szeregowanie zadań o okresach binarnych w systemach silnie uwarunkowanych czasowo.* I Krajowa Konferencja: Metody i systemy komputerowe w badaniach naukowych i projektowaniu inżynierskim, Kraków, 1997, ss. 669-676.
10. R. Nègre: *EUROPRO: A new open VMEbus multiprocessor computer for signal processing and intensive data processing.* Real-Time Magazine, No. 1/1997, pp. 16-20.
11. M. Gajer: *Jak szybko może liczyć komputer?* Wiadomości Elektrotechniczne, nr 12/2002, ss. 519-521
12. A. Kowalczuk: *Ewolucyjna optymalizacja wielokryterialna w automatyce i diagnostyce* AUTOMATION 2003, Konferencja Naukowo-Techniczna, Warszawa, 2-4 kwietnia 2003, ss. 23-38.
13. J. Werewka: *Zagadnienia alokacji zadań w rozproszonych systemach komputerowych czasu rzeczywistego* Elektrotechnika - Kwartalnik AGH, Tom 14, Zeszyt 4, 1995, ss. 479-488
14. O. Kwon, K. Y. Chwa: *Approximation algorithms for general parallel task scheduling*, Information Processing Letters 81 (2002), pp. 143-150.
15. C. L. Liu, J. W. Layland: *Scheduling algorithms for multiprogramming in a hard real time environment* Journal of Association of Computing Machines, vol. 20, No 1, 1973, pp. 46-61.

M. GAJER

SCHEDULING THE SET OF INDEPENDENT, PRE-EMPTIVE AND PERIODIC UNI- AND BIPROCESSOR TASKS WITH THE APPLICATION OF THE METHOD OF SUPER TASKS

Summary

The popularity and ubiquity of real-time systems with hard real-time constraints forced the extensive development of task scheduling theory. In the case of real-time systems with hard real-time constraints it does not suffice that the task produces logically correct results but these results must be delivered within their time constraints. In such systems even logically correct results but delivered with the violation of their time constraints are totally useless. Moreover, the consequences of violation of time constraints can very often be quite severe and can cause the great economic losses and even losses of human lives, e.g. in the case of control systems of nuclear reactors, space ships etc. The main goal of the task scheduling theory is to prove at the stage of the system project that the time constraints for all tasks will always be met under any possible circumstances. In the case of the real-time systems with hard real-time constraints there is very often a necessity of scheduling a set of independent, pre-emptive and periodic tasks. The most popular algorithm for scheduling such set of independent, pre-emptive and periodic tasks is the Rate Monotonic Scheduling algorithm. In the case of Rate Monotonic Scheduling each task is assigned a priority. There are several rules basing on which the priorities are assigned to the tasks and then the tasks are being scheduled. First of all, the shorter the period of task is the higher priority it is assigned. Then, in a given moment, among all the tasks actually in a ready state the one is being executed that has the highest priority. If some task with higher priority enters into the ready state the task being executed

is automatically pre-empted and the task with higher priority begins its execution. The pre-empted task can restart its execution only in the case if there is actually no other task with higher priority in the ready state. The Rate Monotonic Scheduling in its classical form is adequate only for scheduling uniprocessor tasks. This author has propsed a new method of adaptation of Rate Monotonic Scheduling in such a way that it should also be adequate for scheduling the sets of uni- and biprocessor tasks. The clue of the method proposed by this author is concatenation of uniprocessor tasks that form the so called biprocessor super tasks. In order to achieve this the periods of some tasks must be transformed, i.e. they must be shortened in such a way that a subset of uniprocessor tasks could be made. Then each subset of tasks is treated as a uniprocesor task and for the set of such tasks (called by this author super tasks) the Rate Monotonic Scheduling algorithm can be used directly. The method developed by this author was illustrated on the example of scheduling set of tasks for two arbitrary processors. The method can be easily extended both for the case of greater number of processors and for the systems with dedicated processors. For the purpose of finding the suboptimal schedule the use of genetic algorithm was proposed. The method of coding the solution on the genetic material was illustrated on the example.

Keywords: Task Scheduling Theory, Rate Monotonic Scheduling, Multiprocessor Tasks

KWARTALNIK ELEKTRONIKI I TELEKOMUNIKACJI, 2006, 52, z. 1, ss. 89–113

Środowisko APSI wspomagające prototypowanie heterogeniczne modułów zawierających układy FPGA

ERNEST JAMRO [1,2], KAZIMIERZ WIATR [1,2]

[1] *AkademiaGórniczo-Hutnicza, Katedra Elektroniki,*
al. Mickiewicza 30, 30-059 Kraków
[2] *Akademickie Centrum Komputerowe CYFRONET AGH,*
ul. Nawojki 11, 30-950 Kraków
e-mail: jamro@agh.edu.pl, wiatr@agh.edu.pl

Otrzymano 2005.01.05
Autoryzowano 2005.05.04

Niniejszy artykuł opisuje system APSI (ang. *Advanced Programmable Systems Interface*) wspomagający projektowanie i uruchamianie modułów sprzętowych zawierających układ programowalny FPGA. Moduł sprzętowy jest kontrolowany za pomocą komputera PC oraz odpowiedniego środowiska programowego, przez co konieczne stało się zastosowanie heterogenicznego podejścia podczas projektowania, symulacji i testowania. Omawiany system składa się z czterech części: programowej, sprzętowej, symulacyjnej oraz testującej. W ramach części programowej zaproponowano dedykowany język skrypt APSI oraz jego interpreter ułatwiający komunikowanie się z poziomu komputera PC z modułem sprzętowym. W ramach części sprzętowej zaprojektowano moduły sprzętowe, napisane głównie w języku opisu sprzętu VHDL, umożliwiające łatwe komunikowanie się z innymi modułami kompatybilnymi z magistralą Wishbone lub magistralą OPB i środowiskiem EDK. W ramach części symulacyjnej zaproponowano procedurę symulacji heterogenicznej, umożliwiającą łatwą kosymulację dwóch niezależnych wspomnianych powyżej platform: programowej i sprzętowej. Aby umożliwić łatwe testowanie uruchamianych projektów sprzętowych zaprojektowano wewnętrzny analizator stanów logicznych LA_RCS, który umożliwia rejestrowanie i wizualizację przebiegów sygnałów wewnątrz układu FPGA. Środowisko APSI stanowi kompletny system zaproponowany i zaprojektowany w całości przez autorów niniejszej pracy. Zaproponowane oryginalne rozwiązania mogą stanowić podstawę do budowy lub modyfikacji podobnych systemów.

Słowa kluczowe: rekonfigurowalne systemy obliczeniowe, układy programowalne, FPGA, projektowanie układów cyfrowych, Hardware/Software CoDesign

1. WSTĘP

W ostatnich latach gwałtowny rozwój układów programowalnych FPGA (ang. *Field Programmable Gate Array*)[1] spowodował, że znajdują one coraz szersze zastosowanie i coraz częściej stają się one jądrem platformy sprzętowej, dla której przynajmniej w procesie uruchomieniowym bardzo ważna jest komunikacja z komputerem PC. Ponadto układy FPGA są coraz częściej stosowane do akceleracji obliczeń [2], dla których komunikacja z komputerem PC jest kluczowym zagadnieniem.

Głównym wyzwaniem przy integracji heterogenicznych systemów programowo-sprzętowych jest brak odpowiedniego interface'u umożliwiającego łatwe łączenie tych dwóch niezależnych platform oraz ich wzajemnej kosymulacji. Powstało wiele systemów do prototypowania [3,4] czy kosymulacji programowo-sprzętowej [5, 6]. Powyższe systemy nie są jednak kompletne, ponieważ nie zawierają portu komunikacji z komputerem PC, dedykowanego i przyjaznego użytkownikowi interface'u, heterogenicznej kosymulacji, projektowania modułowego lub też wewnętrznego analizatora stanów logicznych.

Środowisko APSI ma wbudowany dedykowany język skryptu, dzięki czemu wymiana danych oraz odczytywanie stanu platformy sprzętowej jest znacznie prostsze. Dla przykładu użycie pojedynczej instrukcji *writeblock* umożliwia łatwy transfer całego pliku lub jego części z komputera PC do określonej lokacji adresowej na platformie sprzętowej.

Środowisko APSI zostało sprawdzone dla płyty XSV firmy Xess [7] podłączonej do komputera PC za pomocą portu równoległego pracującego w trybie EPP (ang. *Enhanced Parallel Port*) – standard IEEE-1284. Jednakże istnieje prosta możliwość modyfikacji środowiska APSI tak, aby komunikował się on przez inne porty, np. PCI, UART czy też USB. Również środowisko APSI może być zastosowane dla innej platformy sprzętowej niż płyta XSV.

Dla części sprzętowej środowisko APSI używa podejścia budowy modułowej. Zostały zaprojektowane specjalne moduły mostków (ang. *bridge*) pomiędzy portem równoległym a magistralą Wishbone [8] lub magistralą OPB (ang. *On-chip Peripherial Bus*) [9] firmy IBM i środowiskiem EDK (ang. *Embedded Development Kit*) firmy Xilinx. Środowisko EDK zostało zaprojektowanie z myślą o modułowej budowie platformy sprzętowej dzięki czemu integracja systemu jest ułatwiona.

Środowisko APSI dodatkowo umożliwia kosymulację heterogeniczną, dzięki czemu w łatwy i szybki sposób można przeprowadzić symulację działania platformy sprzętowej w połączeniu z komputerem PC. Warto w tym miejscu podkreślić, że w literaturze bardzo często stosuje się określenie kosymulacji heterogenicznej dla określenia symulacji działania procesora (wraz z oprogramowaniem) znajdującego się wewnątrz układu FPGA lub jego sąsiedztwie oraz innych modułów sprzętowych znajdujących się wewnątrz (lub w sąsiedztwie) tego samego układu FPGA. W niniejszym artykule część sprzętowa jest rozumiana jako cały układ FPGA wraz z przyległymi modułami sprzętowymi i ewentualnymi wewnętrznymi procesorami oraz ich oprogramowaniem,

a część programowa jako komputer PC wraz ze środowiskiem programowym. W środowisku APSI możliwe jest modelowanie działania środowiska programowego APSI (komputera PC) podczas symulacji sprzętowej odbywającej się na poziomie języka VHDL, jak i śledzenie działania części programowej (platformy PC) na podstawie modelu (wyniku uprzednio przeprowadzonej symulacji VHDL) platformy sprzętowej. Dzięki temu w łatwy sposób można wykryć błędy powstałe podczas procesu projektowania zarówno po stronie sprzętowej jak i programowej. Użytkownik może w prosty sposób zmienić tryb pracy środowiska APSI pomiędzy trybem wykonawczym, podczas którego interpreter skryptu komunikuje się bezpośrednio z platformą sprzętową, a trybem symulacyjnym, podczas którego interpreter skryptu komunikuje się z symulatorem platformy sprzętowej. Kosymulacja heterogeniczna jest bardzo ważną cechą proponowanego systemu ponieważ komputer PC bardzo często odgrywa wiodącą rolę podczas symulacji platformy sprzętowej i vice-versa.

Podobne środowiska zostały uprzednio zbudowane, ale nie są one systemami kompletnymi. Większość systemów koncentruje się na kosymulacji programowo-sprzętowej [4,5,6] ale zakładają one, że kosymulacja odbywa się na tej samej platformie (w tym samym układzie scalonym). Środowisko APSI kosymuluje dwa zupełnie niezależne środowiska: komputer PC oraz platformę sprzętową. Warto podkreślić, że podczas symulacji interpreter skryptu jest uruchamiany bezpośrednio na komputerze PC a nie jak to ma zazwyczaj miejsce w symulatorze sprzętu. Dlatego głównym zagadnieniem jest odpowiednia komunikacja pomiędzy tymi dwoma środowiskami podczas symulacji.

Proces symulacji jest bardzo ważnym etapem prototypowania, jednakże bardzo często trwa on zbyt długo (jest bardzo czasochłonny), nie wszystkie modele symulacyjne są dostępne lub też nie są w pełni zgodne z rzeczywistością, np. stosuje się modele uproszczone, rzeczywiste czasy propagacji są krótsze lub dłuższe niż podczas symulacji, powstają hazardy lub wyścigi, które nie są możliwe do zaobserwowania podczas symulacji, symulacja czasowa nie jest w ogóle przeprowadzana z powodu jej czasochłonności lub braku poprawnych modeli czasowych, lub też pewne elementy elektroniczne są wadliwe. W rezultacie rzeczywisty układ nie działa poprawnie. W takich sytuacjach często jedynym rozwiązaniem jest zastosowanie analizatora stanów logicznych. Do niedawna zewnętrzny analizator stanów logicznych był jedynym rozwiązaniem, jednakże wraz ze wzrostem zasobów układów FPGA pojawiła się możliwość implementacji wewnętrznego analizatora stanów logicznych, który jest implementowany wewnątrz tego samego układu FPGA co układ testowany. Zewnętrzny analizator ma szereg wad w porównaniu z analizatorem wewnętrznym, np. może śledzić tylko zewnętrzne wyprowadzenia, dlatego w celu śledzenia sygnałów wewnętrznych należy je wyprowadzić na zewnątrz co wymaga dokonywania częstych zmian w projekcie układu FPGA oraz ogranicza liczbę równocześnie oglądanych sygnałów, np. do 10-20 [3].

Firma Xilinx udostępnia wewnętrzny analizator stanów logicznych o nazwie ChipScope [10], który jest produktem komercyjnym i używa własnego portu komunikacyjnego oraz środowiska graficznego. Pociąga to za sobą pewne konsekwencje: używa-

ny port komunikacyjny może nie być dostępny na danej platformie sprzętowej oraz moment wyzwolenia może nie być łatwy do zsynchronizowania ze środowiskiem programowym. W konsekwencji zaprojektowano własny wewnętrzny analizator stanów logicznych o nazwie LA_RCS (ang. *Logic Analyzer for Reconfigurable Computing Systems*), który jest bezpośrednio zintegrowany ze środowiskiem APSI oraz EDK i magistralą OPB (albo magistralą Wishbone). Moduł LA_RCS ma szereg zalet w porównaniu z modułem ChipScope, np. używa symulatora VHDL w celu wizualizacji zarejestrowanych sygnałów, przez co zarejestrowane sygnały mogą być dalej łatwo przetwarzane w symulatorze, np. porównywane z wynikiem symulacji lub też wykorzystywane jako wektory wymuszeń. Ponadto moduł LA_RCS zawiera wewnętrzną kompresje danych przez co liczba rejestrowanych próbek została znacząco zwiększona, co ma duże znaczenie ponieważ wielkość wewnętrznej pamięci układu FPGA jest bardzo ograniczona. Ponadto zastosowano zaawansowaną logikę zezwolenia zegara, przez co rejestrowane są tylko wybrane próbki, np. próbkowane są tylko aktywne stany magistrali.

2. ŚRODOWISKO PROGRAMOWE APSI

Wraz ze płytą XSV zostało dostarczone oprogramowanie umożliwiające konfigurację układu FPGA oraz transfer danych poprzez port równoległy komputera PC. Główną wadą tego oprogramowania jest jego skromność zarówno od strony komputera PC jak i od strony źródeł konfigurujących układ FPGA. Wady te były początkową inspiracją do budowy całkowicie nowego i niezależnego środowiska służącego do komunikowania się z płytą XSV z poziomu komputera PC. W związku z tym zaproponowano specjalny język skryptu APSI, który umożliwia łatwą komunikację z platformą sprzętową. Wykonano również interpreter skryptu, program o nazwie roboczej *apsi.exe*, który czyta i wykonuje kolejne komendy zawarte w skrypcie.

Przytaczanie wszystkich komend zaimplementowanych w programie *apsi.exe* wykracza poza ramy niniejszego opracowania, dlatego zostaną wymienione tylko najważniejsze komendy, które dobrze obrazują prezentowane środowisko APSI:

config *nazwa_pliku* – powoduje zaprogramowanie układu FPGA podanym plikiem konfiguracyjnym,

readblock *nazwa_pliku adr_start adr_stop* – powoduje odczytanie zawartości pamięci od adresu *adr_start* do adresu *adr_stop* do podanego pliku,

readbyte *address* – powoduje odczytanie pojedynczej danej spod wskazanej lokacji adresowej, odczytana wartość jest następnie wyświetlona na ekranie oraz zapisywana w rejestrze statusowym,

sleep *czas_ms* – podaje czas wstrzymania wykonywania programu w ms,

run *komenda* – powoduje uruchomienie komendy w linii poleceń systemu operacyjnego, w szczególności możliwe jest uruchomienie innego skryptu poprzez komendę: *run apsi.exe nazwa_skryptu*,

loop *liczba_wykonan* etykieta – powoduje przejście *liczba_wykonan* razy do etykiety *etykieta*,

waitbit0 *maska_bitowa adres* – powoduje czekanie do momentu, aż dana odczytana spod adresu *adres* po dokonaniu operacji bitowej AND z maską *maska_bitowa* będzie równa zeru,

go> *wartosc etykieta* – instrukcja skoku do etykiety jeżeli wartość rejestru statusowego programu APSI jest większa od *wartosc*,

stat+= *wartosc* – powoduje dodanie do rejestru statusowego zmienną *wartosc*,

stat=>m *index* - powoduje zapisanie aktualnej wartości rejestru statusowego w pamięci wewnętrznej programu pod adresem *index*,

filecomp *plik1 plik2* – porównuje dwa pliki.

Podsumowując, komendy APSI umożliwiają wygodne i łatwe komunikowanie się z poziomu komputera PC z platformą sprzętową. Umożliwiają one np. konfigurację układu FPGA oraz transfer danych do/z pamięci. Opracowane środowisko APSI posiada także szereg komend, które poprawiają sterowanie programem np. wykonywanie pętli, uśpienie wykonywania programu. Warto zwrócić uwagę na komendę *waitbit*, która powoduje zatrzymanie wykonywania programu aż do wyzerowania/ustawienia wybranego bitu. Komenda ta ma szerokie zastosowanie np. do odczytu/zapisu pamięci w zależności od stanu gotowości urządzenia. Inną przydatną komendą jest komenda *filecomp*, która porównuje zawartość dwóch plików. Komenda ta umożliwia sprawdzenie czy dane przetworzone przez platformę sprzętową są zgodne z założeniami.

Warto podkreślić, że w niektórych przypadkach możliwości skryptu APSI mogą być niewystarczające i konieczne staje użycie bardziej zaawansowanego języka programowania. Program *apsi.exe* został napisany w języku C++ i składa się z dwóch klas: jednej odwołującej się do platformy sprzętowej na niskim poziomie (klasa *LowLevel*) oraz drugiej interpretującej zawartość skryptu oraz wykonującej operacje na wyższym poziomie. Dlatego w przypadku, gdy instrukcje skryptu są niewystarczające, w sposób naturalny można bezpośrednio odwoływać się do platformy sprzętowej z poziomu języka C++, wykorzystując klasę *LowLevel*. Ponadto skrypt umożliwia uruchomienie programu użytkownika, dzięki czemu dodatkowy program może wykonywać dodatkowe niezawarte w skrypcie czynności.

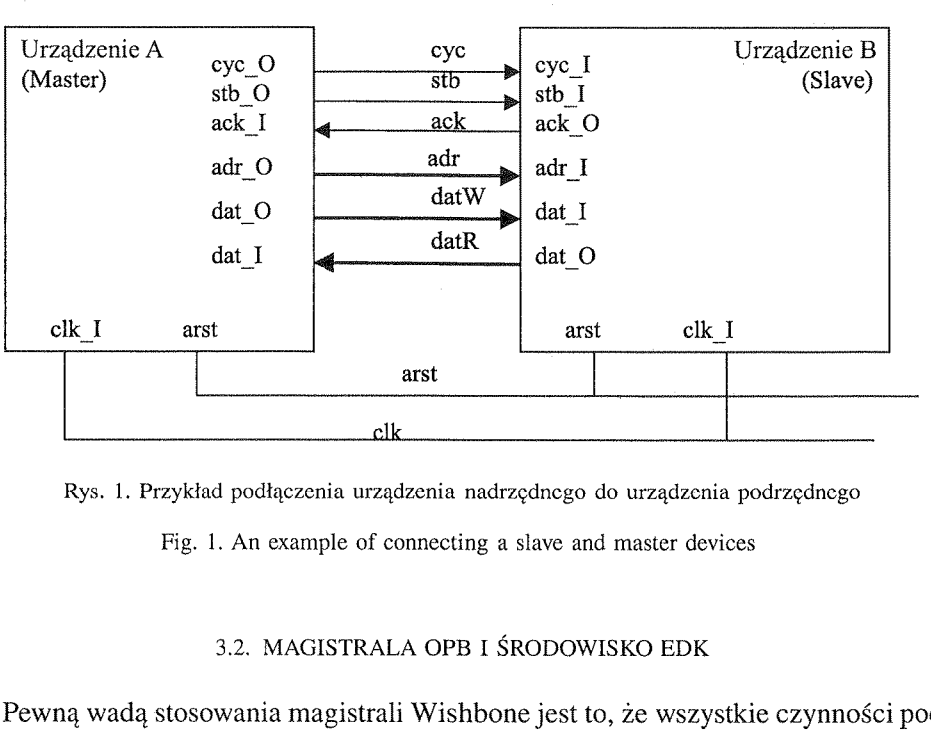
3. PLATFORMA SPRZĘTOWA

Często główną wadą wykonywanych projektów sprzętowych jest brak budowy modułowej i standaryzacji interface'ów poszczególnych modułów. Podobnie było w przypadku firmowego oprogramowania dostarczonego wraz z płytą XSV. Budowa modułowa umożliwia łatwą integrację systemu, śledzenie błędów oraz zmianę i rozbudowę systemu sprzętowego.

3.1. MAGISTRALA WISHBONE

W konsekwencji powyższego wniosku celowe stało się wybranie odpowiedniego standardu magistrali łączącej poszczególne moduły. Wybór padł na początku na magistralę Wishbone, która została zaprojektowana z myślą o łączeniu modułów w projektowaniu SoC (ang. *System – on – Chip*). Jednym z argumentów za wyborem tej magistrali było to, że organizacja OpenCores [8] udostępnia szereg gotowych modułów kompatybilnych z magistralą Wishbone. Moduły te są ogólnie i bezpłatnie dostępne – są wykonywane w ramach GPL (ang. *General Public Licence*), sposób ich udostępniania i wykonywania jest podobny jak w przypadku programów GNU (np. Linux). Wśród dostępnych modułów kompatybilnych z magistralą Wishbone są np. kontroler I^2C, USB, UART, Ethernet. Moduły te zostały tutaj wymienione ze względu na ich występowanie na płycie XSV.

Dzięki zastosowaniu magistrali Wishbone stosunkowo łatwo można łączyć poszczególne moduły: wystarczy tylko odpowiednio podłączyć poszczególne sygnały magistrali do odpowiadających im sygnałów modułu. Można tego dokonać głównie w edytorze tekstu języka VHDL. Przykład podłączenia urządzenia nadrzędnego do urządzenia podrzędnego ilustruje Rys. 1.



Rys. 1. Przykład podłączenia urządzenia nadrzędnego do urządzenia podrzędnego

Fig. 1. An example of connecting a slave and master devices

3.2. MAGISTRALA OPB I ŚRODOWISKO EDK

Pewną wadą stosowania magistrali Wishbone jest to, że wszystkie czynności podłączenia modułów należy dokonywać ręcznie np. w języku VHDL. Czynność ta, chociaż stosunkowo prosta, staje się uciążliwa przy bardziej skomplikowanych projektach oraz przy częstych zmianach i modyfikacjach.

Firma Xilinx zaproponowała własne środowisko EDK (ang. *Embedded Development Kit*), które umożliwia szybkie łączenie gotowych modułów w ramach magistrali OPB (ang. *On-chip Peripherial Bus*) zaproponowanej przez firmę IBM. Magistrala OPB oraz Wishbone są funkcjonalnie podobne, jednakże środowisko EDK przyspiesza łączenie różnych urządzeń poprzez automatyczną generację kodu VHDL i dlatego efektywniejsze jest stosowanie magistrali OPB. EDK wymaga tylko podłączenia urządzenia do magistrali OPB tak jakby to był pojedynczy przewód (można tego dokonać w trybie tekstowym lub graficznym), a poszczególne sygnały magistrali są łączone automatycznie. Dzięki takiej metodzie, łączenie poszczególnych bloków odbywa się szybko i jest czytelne.

W ramach organizacji OpenCores zaprojektowano wiele mikroprocesorów, jednak ich architektura często nie jest zoptymalizowana pod kątem implementacji w układach FPGA, magistrala mikroprocesora nie jest standardem (np. Wishbone) lub też dostępna dokumentacja lub oprogramowanie pozostawia wiele do życzenia. Z drugiej strony firma Xilinx udostępniła jądro procesora MicroBlaze (nazywane soft-procesorem) [11], który jest najszybszym procesorem do implementacji w zasobach programowalnych, dostępnym dla układów FPGA firmy Xilinx. Jest on dobrze udokumentowany oraz współpracuje z magistralą OPB i w bardzo prosty sposób może być dołączony do projektów w ramach pakietu EDK.

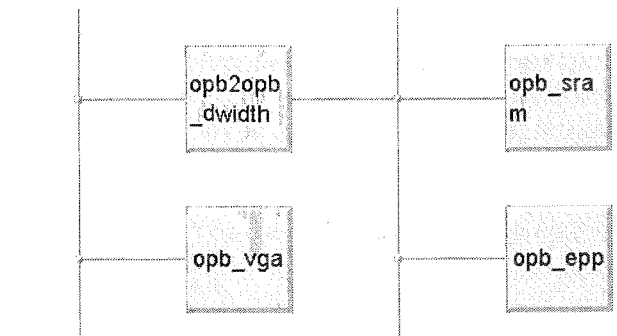
Powyższe argumenty: łatwość łączenia modułów oraz możliwość podłączenia do systemu szybkiego soft-procesora MicroBlaze zadecydowały, że w środowisku APSI zrezygnowano z magistrali Wishbone na rzecz magistrali OPB.

3.3. MODUŁY SPRZĘTOWE

W ramach wykorzystania systemu APSI opracowano szereg modułów umożliwiających łatwe korzystanie z urządzeń dostępnych na płycie XSV. Przykładem takiego modułu jest moduł *opb_epp*. Moduł ten jest mostkiem pomiędzy portem równoległym, a magistralą OPB. Moduł ten oprócz opisu kodu w języku VHDL posiada dodatkowe pliki powodujące, że jest on widoczny w pakiecie EDK. Moduł *opb_epp* można łatwo dołączyć do projektu w EDK podobnie jak to jest w przypadku oryginalnych modułów dostarczonych przez firmę Xilinx w ramach pakietu EDK.

Za pomocą modułu *opb_epp* oraz programu *apsi.exe* można w sposób bardzo prosty zapisywać i odczytywać pamięć zewnętrzną SRAM lub wewnętrzną pamięć BlockRAM układu Virtex, czy też komunikować się z innymi urządzeniami peryferyjnymi podłączonymi do magistrali OPB. Moduł *opb_epp* jest urządzeniem typu nadrzędnego (master) na magistrali OPB dzięki czemu można stworzyć system nie zawierający procesora, lub też moduł *opb_epp* może w pewnym stopniu emulować procesor. Warto w tym miejscu podkreślić, że moduł *opb_epp* jest silnie sparametryzowany dzięki użyciu słowa kluczowego *generic* w języku VHDL. Opis wszystkich parametrów tego modułu wykracza poza ramy niniejszego opracowania.

Oprócz modułu *opb_epp* opracowano również inne moduły, np. moduł *opb_sram* – kontroler pamięci SRAM, czy też interface pomiędzy magistralą OPB a przetwornikiem RAMDAC umożliwiającym wyświetlanie obrazu na monitorze VGA. Przykładowy schemat blokowy projektu wykonanego w pakiecie EDK przedstawia Rys. 2. Projekt ten umożliwia transfer danych pomiędzy komputerem PC a pamięcią zewnętrzną SRAM. Zawartość pamięci SRAM jest wyświetlana na monitorze VGA. Dodatkowy autorski moduł: mostek *opb2opb_dwidth* służy do konwersji szerokości magistrali OPB. Mostek ten posiada również wewnętrzną pamięć FIFO, która umożliwia transfer danych pomiędzy pamięcią SRAM a modułem *opb_epp* w trakcie wyświetlania obrazu. Zastosowanie pamięci FIFO oraz adresowania sekwencyjnego (blokowego) na magistrali OPB zdecydowanie poprawia szybkość transferu danych, tym bardziej że w celu zwiększenia maksymalnej częstotliwości pracy większość modułów wykorzystuje architekturę potokową.



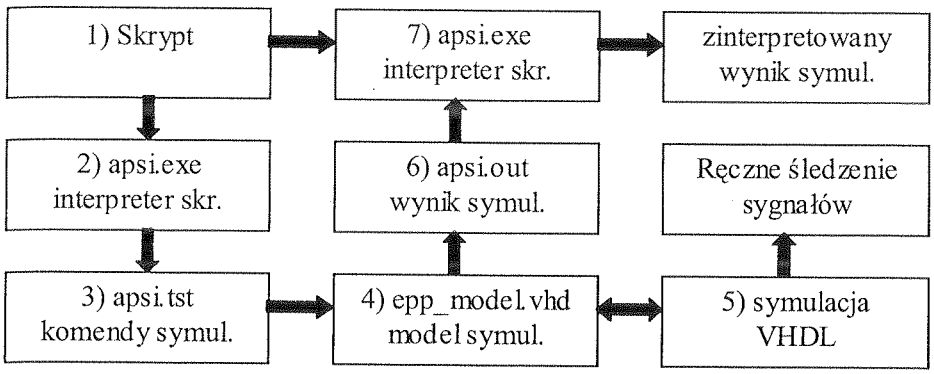
Rys. 2. Przykład projektu w EDK

Fig. 2. An example of EDK project

4. SYMULACJA HETEROGENICZNA

Symulacja styku dwóch różnych architektur jest zadaniem trudnym, a jej brak lub też pobieżność bardzo często pociąga za sobą źle działający finalny projekt. W ramach symulacji środowiska APSI zaproponowano połączenie części software'owej i hardware'owej poprzez następujące składniki:

- interpreter skryptu APSI (program *apsi.exe*), który został odpowiednio zmodyfikowany dla potrzeb symulacyjnych,
- moduł *epp_model.vhd* napisany w języku VHDL, modelujący działanie portu komunikacyjnego (w testowanym przykładzie portu równoległego EPP).



Rys. 3. Etapy symulacji heterogenicznej

Fig. 3. Stages of the heterogeneous simulation

Schemat blokowy kolejnych etapów symulacji heterogenicznej jest podany na Rys. 3. Poszczególne elementy tego schematu zostaną opisane poniżej.

1) Skrypt

Pierwszą czynnością jest określenie komend jakie ma wykonywać program APSI, czyli napisanie odpowiedniego skryptu. Jedyną czynnością dodatkową podczas symulacji heterogenicznej w porównaniu ze standardową pracą systemu APSI jest dodanie na początku skryptu dodatkowej instrukcji *vhdlsim*. Przykładowy skrypt jest przedstawiony na Listing 1, na którym soft-procesor MicroBlaze (schemat blokowy modułu sprzętowego jest przedstawiony na Rys. 4) po wykonaniu zadania ustawia odpowiedni bit semafora. Wykonywanie skryptu jest wstrzymane (instrukcja *waitbit*) do czasu ustawienia tego semafora przez procesor MicroBlaze, po czym wynik obliczeń jest transferowany do komputera PC (*readblock*) i sprawdzany z wynikiem wzorcowym (*filecomp*).

vhdlsim // *program apsi.exe przechodzi w tryb symulacji*
config system // *configuracja ukladu FPGA* (*instrukcja nie symulowana*)
waitbit0 FF 1FF0 // *czekaj na koniec programu* (*komorka semaforu* = 0)
readblock wynik.bin 1F00 1F0F // *czytaj zawartosc zmodyfikowanej pamieci*
filecomp wynik.bin wzor.hex // *porownaj wynik* (*symulacji*) *z wzorem* (*poprawnym wynikiem*)

Listing 1. Przykład skryptu testującego komunikację pomiędzy komputerem a płytą XSV

2) Uruchomienie programu *apsi.exe*

Następnym etapem jest uruchomienie programu *apsi.exe* w celu generacji wymuszeń dla symulatora. Ponieważ w skrypcie umieszczono instrukcje *vhdlsim*, program

apsi.exe nie będzie się komunikował z platformą sprzętową lecz będzie generował wymuszenia dla symulatora, zapisywane w pliku o nazwie roboczej *apsi.tst*.

3) Komendy symulacyjne *apsi.tst*

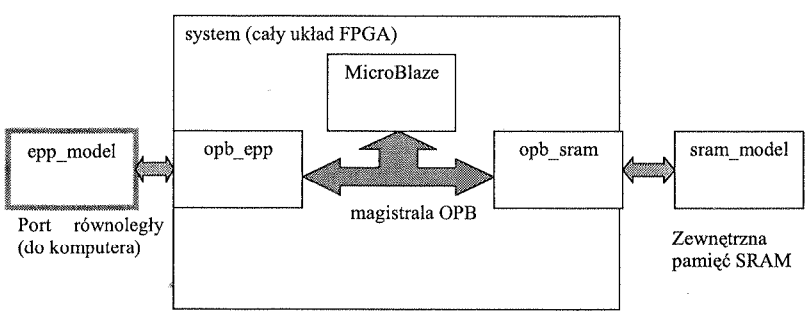
Format pliku binarnego *apsi.tst* służącego do komunikacji pomiędzy interpreterem skryptu (*apsi.exe*) a symulatorem VHDL przedstawiono na listingu 2.

1, a – write address a (byte) – wystawienie na porcie EPP adresu do zapisu, 4 transfery są wymagane aby przesłać jeden adres
2, d – write data d (byte) – wystawienie danej do zapisu na porcie EPP.
3 – read – powoduje odczyt danej z portu EPP
10, t – sleep time – czas wstrzymania aktywności w ms ($t < 256$)
11, b, a_3, a_2, a_1, a_0, m, – waitbit0, waitbit1 – zob. instrukcje skryptu. b=0 waitbit0, b=1 waitbit1, a_i – adres (4 transfery, m- maska)
255 – End Of File – zakończenie, koniec pliku.

Listing 2. Format pliku *apsi.tst*

4) Model symulacyjny portu komunikacyjnego (*epp_model.vhd*)

Następnym krokiem jest przystąpienie do symulacji VHDL. Najpierw należy jednak w projekcie osadzić model symulacyjny portu komunikacyjnego (*epp_model.vhd*). Model ten należy osadzić w nadrzędnym pliku symulacyjnym (testbench) w miejscu gdzie fizycznie znajduje się port komunikacyjny jak to przedstawiono na Rys. 4. Na rysunku tym w podobny sposób osadzono model symulacyjny zewnętrznej pamięci SRAM o nazwie *sram_model.vhd*.



Rys. 4. Schemat blokowy użyty podczas symulacji

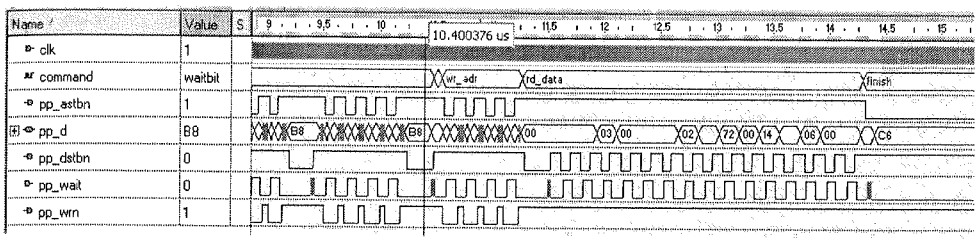
Fig. 4. Simulation block diagram

Moduł *epp_model.vhd* podczas symulacji odczytuje wymuszenia zawarte w pliku *apsi.tst* i na podstawie tych wymuszeń odpowiednio steruje sygnałami portu równole-

głego. Podobny model należy stworzyć w przypadku użycia innego portu komunikacyjnego.

5) Symulacja VHDL

Podczas symulacji właściwej, przeprowadzanej w standardowym symulatorze języka VHDL, możliwe jest śledzenie przebiegów sygnałów wewnętrznych i zewnętrznych w podobny sposób jak to się odbywa podczas standardowej symulacji. Przykład przebiegów na porcie równoległym oraz wykonywanych komend symulacyjnych przedstawia Rys. 5.



Rys. 5. Przykład przebiegów zarejestrowanych na porcie równoległym

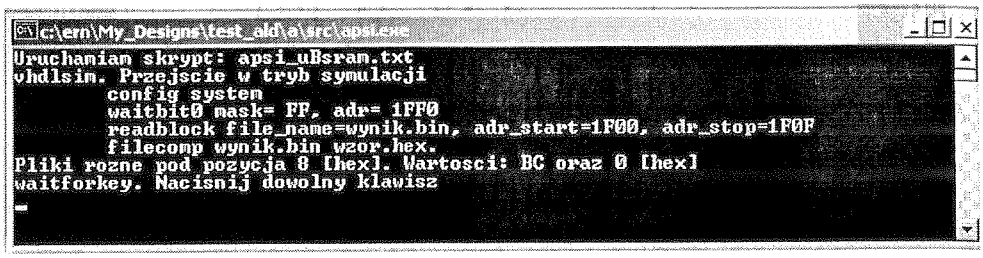
Fig. 5. An example of the simulation waveform observed on the Parallel Port

6) Wynik symulacji (*apsi.out*)

Wynik symulacji można analizować bezpośrednio podczas symulacji VHDL, jednakże bardzo często skomplikowanie przebiegów czasowych praktycznie uniemożliwia sprawdzenie poprawności działania układu. Można to zauważyć analizując Rys. 5. Dlatego dodatkowo podczas symulacji wszystkie dane odczytane z portu równoległego są zapisywane przez model symulacyjny *epp_model.vhd* do specjalnego pliku *apsi.out*.

7) Powtórne uruchomienie programu *apsi.exe*

Następną czynnością jest ponowne uruchomienie interpretera skryptu. Tym razem interpreter dysponuje wynikiem symulacji zapisanym w pliku *apsi.out* i na podstawie tego pliku zachowuje się tak, jakby dane zapisane w tym pliku zostały w rzeczywistości odczytane z portu równoległego. W konsekwencji interpreter skryptu zachowuje się tak, jakby w rzeczywistości kontaktował się z płytą XSV podłączoną do portu komunikacyjnego. Przykładowy wynik uruchomienia programu *apsi.exe* dla skryptu przedstawionego w Listing 1 jest pokazany na Rys. 6. W wyniku przeprowadzonej symulacji heterogenicznej pokazano, że ósmy bajt wyniku jest błędny.



Rys. 6. Wynik uruchomienia programu apsi.exe po wykonaniu symulacji heterogenicznej

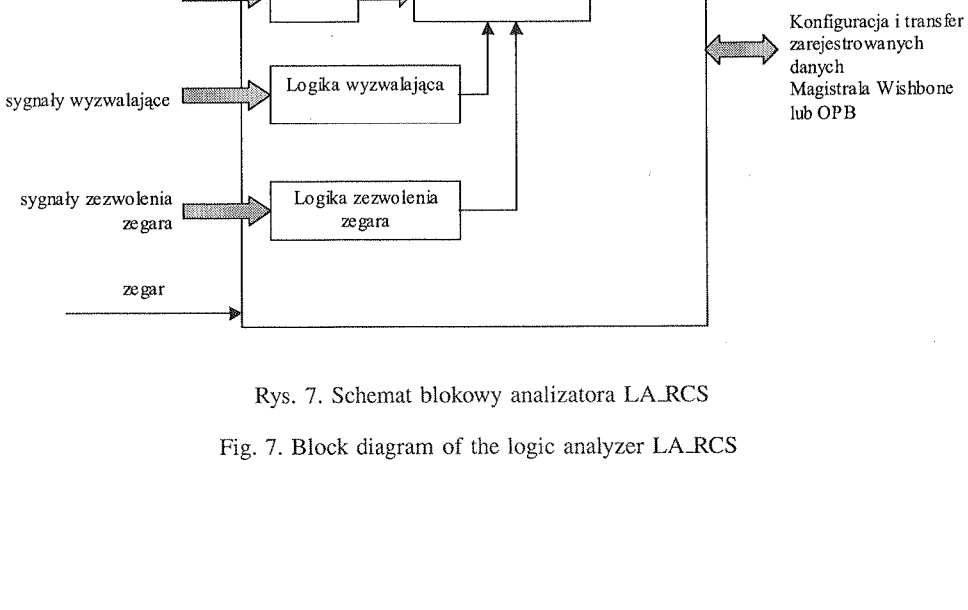
Fig. 6. Heterogeneous simulation results generated by the program apsi.exe

5. WEWNĘTRZNY ANALIZATOR STANÓW LOGICZNYCH

Symulacja jest bardzo ważnym etapem projektowania, jednakże bardzo często pomimo poprawnego wyniku symulacji rzeczywisty układ nie działa poprawnie. W takim przypadku najlepszym rozwiązaniem staje się użycie analizatora stanów logicznych, w szczególności wewnętrznego analizatora stanów logicznych LA_RCS.

5.1. BUDOWA ANALIZATORA LA_RCS

Ogólna koncepcja działania wewnętrznego analizatora stanów logicznych LA_RCS jest bardzo podobna jak w przypadku analizatorów zewnętrznych. Analizator LA_RCS w przybliżeniu składa się z następujących modułów:



Rys. 7. Schemat blokowy analizatora LA_RCS

Fig. 7. Block diagram of the logic analyzer LA_RCS

- Moduł próbkujący dane.
- Moduł logiki wyzwalającej (ang. *trigger*).
- Modułu zezwolenia zegara CE (ang. *ClockEnable*).
- Moduł kompresji rejestrowanych próbek.
- Pamięć rejestrowanych próbek.
- Moduł konfigurujący i transferujący zarejestrowane dane.
- Moduł wyświetlający zarejestrowane dane.

5.2. MODUŁ PRÓBKUJĄCY DANE

Podobnie jak w przypadku analizatora zewnętrznego pierwszą czynnością jest przyłączenie sygnałów, które mają być obserwowane. W przypadku analizatora zewnętrznego należy podłączyć sondy, używając odpowiednich końcówek, które następnie transferują dane do analizatora. W przypadku analizatora wewnętrznego podpięcie sond polega na umieszczeniu modułu analizatora w projekcie oraz doprowadzenie na odpowiednie wejścia sygnałów, które maja być obserwowane. Dodatkową czynnością jest odpowiednie podłączenie magistrali konfigurującej i transferującej dane do komputera. Istnieją dwie opcje analizatora i odpowiadające im dwa alternatywne sposoby podłączenia analizatora:

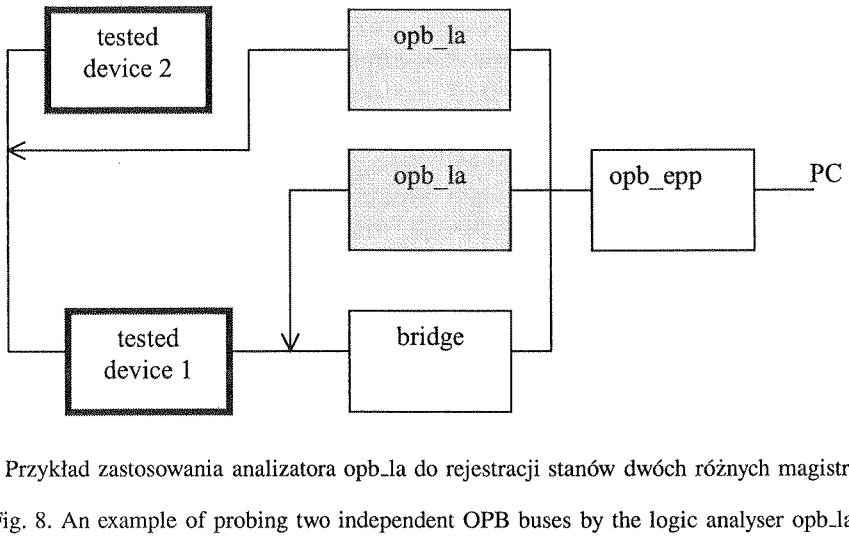
1) opcja VHDL – przyłączenie analizatora bezpośrednio w kodzie języka VHDL, w tym przypadku należy użyć modułu *log_anal.vhd*, który komunikuje się z komputerem PC poprzez magistralę Wishbone. Przykład osadzenia analizatora pokazuje Listing 3.

```
la: log_anal
generic map (data_width=> 32, -- liczba rejestrowanych sygnałów
mem_adr_width=> 9, -- liczba rejestrowanych próbek (512)
adr_width=> 12, -- szerokość magistrali adresowej konfigurującej
trig_width=> 32, -- liczba sygnałów wejściowych wyzwalających
ce_dwidth=> 32; -- liczba sygnałów wejściowych logiki CE lub drugiego triggera
two_clocks=> 0, -- niezależny sygnał zegarowy kontrolny i rejestrujący
use_run_length_coding=> 1) -- użycie kompresji rejestrowanych danych
port map (arst=> arst, -- asynchroniczny reset
clk=> la_clk, -- sygnał zegarowy rejestrujący dane
-- interface części rejestrującej dane
data=> la_data, -- dane, które mają być rejestrowane
ce_data=> la_ce_data, -- magistrala zezwolenia zegara dla rejestrowanych danych
trig=> la_trig, -- sygnały wejściowe logiki wyzwalającej
ce_trig=> la_ce_trig, -- zezwolenie zegara dla logiki wyzwalającej
-- interface kontrolny magistrali wishbone
wb_clk_I=> wb_clk, -- sygnał zegarowy magistrali wishbone
wb_adr_I=> wb_adr(11 downto 0), -- magistrala adresowa
```

```
wb_dat_I=> wb_datW, -- dane do zapisu
wb_dat_O=> wb_datR, -- dane do odczytu
wb_stb_I=> wb_stb, -- urządzenie master inicjalizuje transfer danych
wb_we_I=> epp_we, -- kierunek transferu (zapis / odczyt)
wb_ack_O=> la_ack); -- urządzenie slave gotowe do transferu
```

Listing 3. Przykład podłączenia analizatora w układzie

2) opcja EDK – przyłączenie analizatora bezpośrednio w środowisku EDK przy pomocy modułu *opb_la*. Moduł *opb_la* posiada dwie niezależne magistrale OPB: pierwsza z nich służąca do transferu danych i konfiguracji poprzez komputer PC; druga służąca do próbkowania stanu badanej magistrali OPB (dodatkowe sygnały zewnętrzne umożliwiają badanie innych sygnałów niż magistrala OPB). Przykład podłączenia dwóch niezależnych modułów *opb_la* do rejestracji stanów dwóch niezależnych magistral OPB pokazuje Rys. 8.



Rys. 8. Przykład zastosowania analizatora opb_la do rejestracji stanów dwóch różnych magistral OPB

Fig. 8. An example of probing two independent OPB buses by the logic analyser opb_la

Pewną wadą opisywanego narzędzia jest to, że aby zmienić rejestrowane sygnały na inne wymagana jest zmiana w projekcie oraz powtórna konfiguracja układu FPGA. W przypadku, kiedy powyższa czynność stanowi duże ograniczenie można użyć większej szerokości danych analizatora, co jednak zwiększa zapotrzebowanie na pamięć. Alternatywnym i bardziej zalecanym rozwiązaniem jest użycie dodatkowych multiplekserów, które w zależności od potrzeb przyłączają do analizatora sygnały, które mają być oglądane. Wymaga to jednak użycia dodatkowej logiki przełączającej.

Rejestrowane sygnały są taktowane zewnętrznym sygnałem zegarowym, analizator nie posiada wewnętrznego sygnału zegarowego. Ponadto dane są zatrzaskiwane tylko dla narastającego sygnału zegarowego, czyli tylko raz na okres zegara. W przypadku

jednak gdy jest potrzeba rejestracji więcej niż jednej próbki na okres zegara możliwe jest zwielokrotnienie częstotliwości zegara np. w bloku DLL (ang. *DelayLockLoop*).

5.3. MODUŁ LOGIKI WYZWALAJĄCEJ

Analizator zewnętrzny bardzo często ma ograniczoną liczbę sond wejściowych, dlatego sygnały rejestrowane oraz sygnały wyzwalające są takie same. W przypadku analizatora wewnętrznego liczba sygnałów wejściowych nie ma dużego znaczenia i dlatego użyto niezależnych wejść dla rejestrowanych danych oraz dla wejść logiki wyzwalającej, co daje większe możliwości, np. sygnały wejściowe logiki wyzwalającej nie muszą być rejestrowane.

Miejsce sygnału wyzwalającego w odniesieniu do rejestrowanych sygnałów również można określić poprzez odpowiedni wpis przez magistralę kontrolną. Konsekwentnie sygnał wyzwalający może być na początku, na końcu lub też w dowolnym miejscu pomiędzy początkiem i końcem rejestrowanego przebiegu. Łatwa do realizacji jest rejestracja przebiegów, dla których sygnał wyzwalający znajduje się na początku rejestrowanego przebiegu. W innych przypadkach proces rejestracji wymaga użycia bardziej skomplikowanej logiki, która wymaga aby sygnały były rejestrowane zawsze niezależnie od tego czy wystąpił sygnał wyzwolenia, zakończenie rejestracji następuje dopiero w odpowiednim czasie po spełnieniu warunku wyzwolenia. Czas ten jest zdefiniowany poprzez miejsce wystąpienia wyzwolenia na tle obserwowanych próbek.

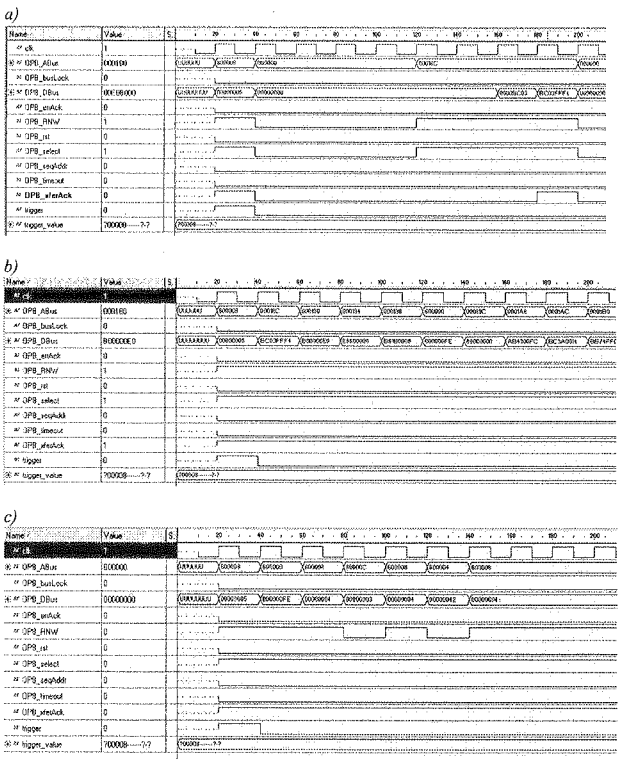
W opisywanym analizatorze użyto bardzo prostej logiki wyzwalającej 01X, która polega na tym że analizator ulega wyzwoleniu w momencie, kiedy stan każdego z wejść ma wartość 0 lub 1 albo też jego wartość nie ma znaczenia (oznaczenie X). Wartość logiki 01X jest określana niezależnie dla każdego sygnału wejściowego poprzez odpowiedni wpis do rejestru poprzez magistralę kontrolną i dlatego nie jest wymagane przeprogramowanie układu FPGA w celu zmiany logiki wyzwalającej. Układ posiada dwa niezależne moduły wyzwalające z logiką 01X. Moduły te są powiązane logiką: $a\ AND\ b$; $a\ OR\ b$; $a\ AND\ NOT\ b, a\ OR\ NOT\ b$ (gdzie a, b – wyjścia dwóch modułów wyzwalających). Dopiero spełnienie powyższej logiki powoduje wyzwolenie analizatora. Ponadto moduł b może być opóźniony o jeden takt zegara względem modułu a, co umożliwia ustawienie wyzwolenia w zależności od zmiany stanów sygnałów, np. kiedy sygnał zmienia się ze stanu 00XX na 100X, w szczególności umożliwia to detekcje zboczy. Wejścia tych dwóch układów generujących wyzwolenie są niezależne.

5.4. MODUŁ ZEZWOLENIA ZEGARA

Dużą zaletą proponowanego analizatora jest dodanie nowatorskiej logiki zezwolenia zegara CE dla rejestrowanych danych. Sygnał CE w przybliżeniu działa podobnie jak dodanie bramki AND na sygnał zegara i sygnał CE, lecz logika CE jest pozbawiona ewentualnych wyścigów. Sygnały CE umożliwiają zatem wybiórczą rejestrację sygna-

łów. Przykładem wykorzystania sygnałów CE jest rejestracja tylko aktywnych stanów na magistrali lub też stanów, w których następuje transfer do określonych urządzeń.

Moduł analizatora posiada logikę generacji sygnału CE podobną jak to miało miejsce w przypadku układu wyzwalającego: użyto logiki 01X oraz dodatkowo magistrali wejściowej, której stan jest wejściem logiki 01X. Konsekwentnie można szybko zmienić momenty czasowe, w których dane są rejestrowane. Rejestrowanie sygnałów można uzależnić od sygnałów kontrolnych magistrali (np. brak rejestracji momentów bezczynności), od adresu, zapisu/odczytu (można rejestrować tylko transfery danych odbierane przez dane urządzenie) itp.



Rys. 9. Przykład rejestracji tego samego przebiegu dla trzech różnych opcji logiki CE: a) bez użycia logiki CE, b) rejestracja tylko cykli aktywnych (OPB_xferAck=1), c) rejestracja tylko przesłań do urządzenia opb_uart_lite

Fig. 9. An example of probing the same OPB bus input sequence with three different clock enable options: a) without clock enable, b) active OPB cycles (OPB_xferAck=1), c) capture only active transfers when only a specified device is addressed

Przykład rejestracji tego samego przebiegu magistrali OPB dla trzech różnych ustawień logiki CE pokazuje Rys. 9. W przebiegu *a* rejestrowane są wszystkie przebiegi (brak logiki CE); w przebiegu *b* rejestrowane są tylko aktywne stany ma magistrali OPB

(OPB_xferAck=1), w przebiegu *c* rejestrowane są tylko aktywne stany magistrali OPB adresujące wybrane urządzenie. Jak widać poszczególne przebiegi niosą porównywalną informację, jednakże są one użyteczne na różnym poziomie testowania. Wydobycie przebiegów aktywnych adresujących wybrane urządzenie na podstawie przebiegu *a* jest bardzo czasochłonne i mało efektywne w porównaniu z przebiegiem *c*. Dlatego użytkownik stosownie do swoich potrzeb może wybrać stosowną opcję logiki CE. Warto podkreślić, że logika CE nie tylko powoduje zmniejszenie liczby rejestrowanych próbek, co ma duże znaczenie w przypadku niewielkich rozmiarów pamięci wewnętrznej układu FPGA, ale również powoduje znaczne zwiększenie czytelności rejestrowanych przebiegów. Szkoda, że do tej pory nie zastosowano podobnej selekcji wyświetlania danych w przypadku symulatorów.

5.5. MODUŁ KOMPRESJI REJESTROWANYCH PRÓBEK

Dużą wadą wewnętrznego analizatora stanów logicznych jest mały rozmiar pamięci jaką z reguły ma do dyspozycji analizator w pojedynczym układzie FPGA. Dlatego dodano dodatkowy moduł prostej kompresji danych: kodowanie długości serii RLC (ang. *Run Length Coding*). Kodowanie RLC polega na tym, że powtarzające się wyrazy są kodowane jako wartość wyrazu oraz liczba powtórzeń. Jako znacznik czy kodowana jest wartość sygnału czy też liczba powtórzeń wykorzystano najstarszy bit. Znacznik równy zero oznacza, że kodowana jest wartość sygnału, znacznik 1 oznacza że kodowana jest liczba powtórzeń poprzedniego sygnału. W konsekwencji liczba rejestrowanych sygnałów jest pomniejszona o jeden bit znacznika. Przykład zakodowania podaje Listing 4.

```
Sygnały wejściowe przed kodowaniem:
    0, 1, 2, 2, 2, 3, 3, 3, 4, 4, 4, 4, 0, 1, 1, 2, ...
Sygnały wyjściowe po kodowaniu:
    0, 1, 2, 81, 3, 82, 4, 83, 0, 1, 2, ...
```

Listing 4. Przykładowy ciąg danych przed i po kodowaniu RLC dla 8-bitowych słów

Warto podkreślić, że zastosowanie kodowania RLC nie powoduje wydłużenia kodowanej sekwencji. Jedyną wadą użycia kodowania RLC jest to, że liczba rejestrowanych sygnałów uległa zmniejszeniu o 1 (bit znacznika). Ponadto dodatkowa powierzchnia zajmowana przez koder RLC jest niewielka: w pierwszym przybliżeniu koder składa się z komparatora, licznika oraz multipleksera *n*-bitowego (gdzie *n* – szerokość rejestrowanych danych).

Użycie kodowania RLC umożliwia znaczące zwiększenie liczby rejestrowanych próbek. Prostym przykładem może być moduł wolnej transmisji szeregowej UART. Rejestracja przesłania pojedynczej danej może wymagać rejestracji milionów próbek z częstotliwością zegara systemowego, natomiast liczba faktycznych czynności może być niewielka. Zastosowanie standardowego analizatora, nawet zewnętrznego o dużej ilości rejestrowanych próbek, może okazać się niewystarczające, natomiast proponowa-

ny analizator spełni stawiane wymagania. Alternatywą dla analizatora bez kompresji RLC może być zmniejszenie częstotliwości próbkowania, jednakże w tym przypadku szybkie przebiegi przejściowe i zakłócenia nie są rejestrowane.

Warto podkreślić, że kodowanie RLC jest na tyle efektywne, że podczas wyświetlania zarejestrowanych próbek konieczne stało się ograniczenie (definiowane przez użytkownika) maksymalnej ilości powtórzeń tak aby wyświetlany przebieg nie został zdominowany przez powtarzające się sygnały (np. bardzo długi czas bezczynności na tle relatywnie krótkich czasów aktywnych).

5.6. PAMIĘĆ REJESTROWANYCH PRÓBEK

Wszystkie rejestrowane próbki są zapisywane do pamięci wewnętrznej i następnie transmitowane *off-line* do komputera PC, gdzie następnie są przetwarzane. Pewnym ograniczeniem stosowania wewnętrznego analizatora stanów logicznych jest niewielka pamięć wewnętrzna dostępna w pojedynczym układzie scalonym FPGA. Dlatego niektóre narzędzia komercyjne np. ILA/ATC (Agilent TPA) [10] stosują pamięć zewnętrzną, wiąże się to jednak z ograniczoną częstotliwością próbkowania lub mniejszą szerokością magistrali danych. Warto jednak podkreślić, że poprzez zastosowanie kompresji danych oraz coraz większej pojemności dostępnej pamięci w jednym układzie scalonym powyższe ograniczenie wielkości pamięci jest coraz mniej uciążliwe. Z praktycznego punktu widzenia liczba próbek 512 wraz z kompresją i wybiórczą rejestracją (użycie zaawansowanej logiki CE) jest w dużej ilości przypadków wystarczająca i większa liczba próbek jest z reguły trudna do prześledzenia. Warto podkreślić, że liczba próbek jest określona parametrem i może być zmieniona przez użytkownika. Dla układu Virtex zalecana liczba próbek to 512 do 4096, a dla układu Virtex II jest ona 4-krotnie większa (co wynika z wielkości bloków pamięci BlockRAM w tych układach FPGA).

5.7. MODUŁ KONFIGURUJĄCY I TRANSFERUJĄCY ZAREJESTROWANE DANE

Konfiguracja modułu analizatora oraz odczyt zarejestrowanych danych odbywa się poprzez magistralę Wishbone [8] lub magistralę On-chip Peripherial Bus (OPB) [9] oraz odpowiednie moduły mostków łączących te magistrale z komputerem PC (zob. podrozdział 3.3).

Warto podkreślić, że moduł analizatora LA_RCS jest bezpośrednio zintegrowany ze środowiskiem APSI i istnieją komendy umożliwiające w łatwy sposób konfigurację analizatora oraz odczyt zarejestrowanych danych. W konsekwencji używając jednego środowiska APSI możliwe jest obsługiwanie całego systemu. Jest to o tyle ważne, że w ten sposób ułatwiona jest synchronizacja czasowa. W szczególności w lepszy sposób można kontrolować badany układ włącznie z wyzwoleniem analizatora poprzez środowisko APSI, co w dużym stopniu upraszcza logikę wyzwalającą. Przykładem może być wyzwolenie analizatora (z poziomu skryptu APSI) tuż przed dokonaniem transferu danych (np. instrukcją *writeblock*), który chcemy obserwować. Czynność ta jest

trudna do uzyskania w przypadku użycia innego narzędzia np. ChipScope [10]. Warto podkreślić, że analizator LA_RCS może być wielokrotnie wykorzystywany podczas pojedynczego uruchomienia skryptu APSI, tj. zarejestrowane próbki mogą być transferowane do komputera (lub pamięci zewnętrznej), a następnie analizator może być powtórnie użyty, przez co liczba rejestrowanych próbek lub też momentów czasowych może być zwielokrotniona.

5.8. MODUŁ WYŚWIETLAJĄCY ZAREJESTROWANE DANE

Moduł wyświetlający zarejestrowane dane o nazwie *la_view.vhd* wykorzystuje dane uprzednio zapisane w formie pliku na komputerze PC. Moduł *la_view* został napisany w języku VHDL i używa symulatora języka VHDL do wyświetlania zarejestrowanych danych. Grupowanie sygnałów oraz nadawanie im nazw odbywa się na poziomie języka VHDL w pliku *la_view.vhd*.

Zastosowane rozwiązanie posiada szereg zalet:

- Uniknięto projektowania skomplikowanego programu interface'u użytkownika oraz użyto znany użytkownikowi (jego własny) interface symulatora języka VHDL.
- Nazywanie i grupowanie sygnałów odbywa się bezpośrednio w języku VHDL z wykorzystaniem jego składni.
- Otrzymane przez analizator sygnały mogą służyć bezpośrednio jako wektor wymuszeń do symulacji danego modułu. Opcja ta może być bardzo użyteczna ponieważ wyręcza projektanta z czasochłonnego wpisywania sygnałów wymuszających, które zostały zarejestrowane w rzeczywistości. Większość analizatorów stanów logicznych posiada swój własny interface i jest pozbawiona tej cechy.
- Analiza lub porównanie zarejestrowanych przez analizator przebiegów z wynikiem symulacji może być przeprowadzona bezpośrednio w symulatorze VHDL.

Warto podkreślić, że w przypadku użycia modułu *opb_la*, użytkownik nie musi edytować nazw sygnałów w pliku *la_view.vhd*. Plik ten posiada predefiniowane połączenia dla magistrali OPB. Konieczne jest tylko podanie parametrów takich jak szerokość magistrali danych i liczba rejestrowanych próbek.

5.9. WYNIK IMPLEMENTACJI

Przedstawienie wyniku implementacji analizatora LA_RCS jest trudne ponieważ moduł ten jest silnie sparametryzowany. Dlatego zostanie pokazany tylko wynik dla jednego zestawu parametrów. Wynik ten podaje Listing 5 dla układu Virtex XCV300PQ240-6.

Parametry:
data_width:= 16 – liczba próbkowanych sygnałów
mem_adr_width:= 9 – szerokość magistrali adresowej pamięci (liczba próbek na sygnał: 512)
trig_width:= 8 – liczba sygnałów układu wyzwalającego

two_clocks:= 0 – pojedynczy sygnał zegarowy (jedna próbka na cykl zegara)
ce_dwidth=> 0; -- brak logiki CE

Wynik implementacji
Number of Slices: 78 out of 3,072 2%
Number of Slice Flip Flops: 40 out of 6,144 1%
Total Number 4 input LUTs: 116 out of 6,144 1%
Number used as LUTs: 105
Number used as a route-thru: 11
Number of Block RAMs: 2 out of 16 12%
Total equivalent gate count for design: 34,245
Minimum period is 13.492ns.

Listing 5. Wynik implementacji układu LA_RCS

Warto podkreślić, że największe zasoby układu FPGA są wykorzystywane przez pamięć próbek czyli przez pamięć blokową BRAM. W większości przykładów można przybliżyć zajmowaną powierzchnię analizatora LA_RCS tylko do liczby zajmowanych pamięci BRAM. Liczbę tę można łatwo oszacować, pokazuje to Tab. 1 dla układu Virtex II.

Tabela 1

Liczba wykorzystanych pamięci blokowych BRAM w zależności od użytych parametrów analizatora LA_RCS (data_width – liczba rejestrowanych kanałów, mem_adr_width – szerokość magistrali adresowej pamięci – liczba próbek na kanał= $2^{mem\_adr\_width}$)

The number of Block RAMs (BRAM) used for different LA_RCS parameters (data_width – the number of captured channels, mem_adr_width – internal memory address width – number of samples is equal $2^{mem\_adr\_wisth}$)

# BRAM	data_width	mem_adr_width
1	8	11
	16	10
	32	9
2	8	12
	16	11
	32	10
	64	9
4	8	13
	16	12
	32	11
	64	10

5.10. PORÓWNANIE ANALIZATORÓW ZEWNĘTRZNYCH I WEWNĘTRZNYCH

Porównując zewnętrzny i wewnętrzny analizator stanów logicznych można dokonać analizy wad obu tych rozwiązań.

Wady analizatora zewnętrznego:

1) Duża liczba obserwowanych sygnałów powoduje, że poprawne podłączenie sondy do odpowiednich miejsc układu jest często czasochłonne i narażone na błędy (np. sygnał został podłączony w nieodpowiednim miejscu lub też została pomylona kolejność podłączonych sygnałów).
   W przypadku analizatora wewnętrznego, podłączenie sygnałów odbywa się w projekcie (np. w opisie VHDL) i przez to jest łatwiejsze do wykonania i przeanalizowania. Dla przykładu dla magistrali 32-bitowej w języku VHDL możemy zastosować proste pojedyncze przypisanie. Wadą analizatora wewnętrznego jest to, że zmiana oglądanych sygnałów wymaga dokonywania zmian w projekcie i powtórnej implementacji całego projektu w układzie FPGA.
2) Bardzo małe rozmiary współczesnych układów scalonych powodują, że trudno jest podłączyć poszczególne sygnały, końcówki sondy analizatora często źle kontaktują, często odpadają lub też zwierają poszczególne ścieżki w testowanym układzie. Warto w tym miejscu podkreślić, że sondy są bardzo często potencjalnym zagrożeniem dla badanego układu, ponieważ łatwo mogą powodować zwarcia poszczególnych części układu prowadząc do jego uszkodzenia. Dlatego zalecane jest podłączanie sond przy wyłączonym zasilaniu – co z kolei powoduje, że zmiana rejestrowanych sygnałów jest bardzo czasochłonna (konieczność każdorazowej konfiguracji układu FPGA). Rozwiązaniem wymienionych problemów jest stosowanie w testowanym układzie złącz przeznaczonych specjalnie dla analizatora.
   Dla analizatora wewnętrznego wspomniany problem nie występuje.
3) Podłączenie sondy powoduje dodatkowe obciążenie badanego układu. W przypadku dużej częstotliwości próbkowanych sygnałów duży wpływ może mieć szczególnie obciążenie o charakterze pojemnościowym. Dla przykładu: dla analizatora stanów logicznych TLA 600 firmy Tektronix sonda stanowi obciążenie rezystancyjne 20 kΩ oraz pojemnościowe 2 pF. Dla tej pojemności czas narastania napięcia od 0 do 2.4 V dla prądu ładującego 20 mA wynosi 0.24 ns. Przy dużych częstotliwościach obserwowanych sygnałów, długość połączeń jest porównywalna z długością fali (np. dla częstotliwości f = 1GHz długość fali jest mniejsza niż 30 cm) przez co połączenia muszą być traktowane jako linie długie, na których powstają odbicia. Dlatego bardzo ważnym zagadnieniem jest odpowiednie prowadzenie i dopasowanie impedancyjne połączeń w badanym układzie. Podłączenie do takich ścieżek sondy powoduje powstawanie niedopuszczalnych zakłóceń. Innym problemem jest różny czas opóźnień analizatora dla różnych rejestrowanych kanałów. Dla przykładu dla analizatora TLA 600, który umożliwia próbkowanie co 0.5ns (2 GSample/s) różnica opóźnień (przesunięcie czasowe) na poszczególnych kanałach może wynosić aż 1.6 ns.

W przypadku analizatora wewnętrznego podłączenie analizatora powoduje dodatkowe obciążenie badanego układu, jednakże jest ono mniejsze (połączenie wewnętrzne), dobrze określone (możliwe do przeanalizowania np. w symulatorze). Ponadto warto dodać, że wszystkie wejścia analizatora LA_RCS są wejściami rejestrowymi (każde wejście jest najpierw zatrzaskiwane w przerzutniku typu D) dzięki czemu wpływ analizatora na wewnętrzne połączenia jest mniejszy oraz analizator działa szybciej.

4) Fundamentalną wadą analizatora zewnętrznego jest to, że umożliwia rejestrowanie tylko wyprowadzeń zewnętrznych. Coraz częściej w jednym układzie scalonym znajduje się cały system – układy SoC (ang. *System on Chip*) i dlatego konieczne jest również obserwowanie co dzieje się w środku układu scalonego. Dlatego w takim przypadku należy użyć analizatora wewnętrznego, który stanowi integralną część układu scalonego i umożliwia obserwowanie wybranych sygnałów wewnętrznych.

Wady analizatora wewnętrznego:

1) Główna wadą jest brak możliwości szybkiej zmiany obserwowanych sygnałów ponieważ analizator jest włączony w proces projektowy. Możliwe jest jednak podłączenie bardzo dużej liczby sygnałów do analizatora poprzez układ multiplekserów, który wybiera ostatecznie, które sygnały mają być rejestrowane. W przypadku układów FPGA istnieje relatywnie szybka możliwość podłączenia innych sygnałów do wejść analizatora, a następnie przeprogramowanie układu FPGA.
   W przypadku analizatora zewnętrznego teoretycznie istnieje łatwa możliwość zmiany oglądanych sygnałów, jednakże trudności wymienione w poprzednich akapitach powodują, że niejednokrotnie łatwiejsze jest przeprogramowanie układu FPGA niż przełączenie sondy analizatora.
2) Analizator wewnętrzny zajmuje dodatkowe zasoby badanego układu scalonego, szczególnie zasoby pamięciowe. Jednakże analizator może być użyty tylko podczas testowania projektu, później układ FPGA może być zaprogramowany projektem nie posiadającym analizatora.
   W przypadku analizatora zewnętrznego, ten sam analizator może być użyty wielokrotnie do przebadania różnych układów. Warto jednak podkreślić, że koszt takiego analizatora jest wielokrotnie większy niż koszt analizatora wewnętrznego.
3) W celu ograniczenia powierzchni zajmowanej przez analizator wewnętrzny, wielkość pamięci oraz logika analizatora (a przez to i możliwości) są dużo mniejsze niż w przypadku (znacznie droższego) analizatora zewnętrznego. Warto podkreślić, że poprzez dodanie kompresji rejestrowanych danych liczba rejestrowanych próbek uległa znacznemu powiększeniu w porównaniu z wielkością pamięci. Ponadto zastosowanie analizatora wewnątrz układu scalonego umożliwia dodanie przez konkretnego użytkownika dodatkowej logiki stosownie do wymagań.

5.11. PORÓWNANIE Z CHIPSCOPE'M

Opisywany analizator LA_RCS jest rozwiązaniem konkurencyjnym w stosunku do produktu ChipScope [10] firmy Xilinx. Jest jednak w stosunku do niego bardziej uproszczony. Słabszą stroną analizatora LA_RCS jest układ wyzwalający, który stosuje tylko logikę 01X i posiada dwa niezależne moduły. Produkt ChipScope posiada wiele niezależnych modułów wyzwalających (do 16), które mogą reagować na zmiany czasowe sygnałów – zbocze narastające, opadające lub zmiana sygnału, oraz umożliwiają porównywanie wartości sygnałów. Ponadto ChipScope posiada niezależny interface do wyświetlania zarejestrowanych przebiegów. Ważną zaletą ChipScope'a jest również to, że jest on zintegrowany z pakietem ISE – programem narzędziowym firmy Xilinx'a do projektowania i programowania układów FPGA. Dzięki temu zmiana oglądanych sygnałów może następować nie tylko na poziomie języka VHDL, jak to ma miejsce w LA_RCS, ale również w późniejszych etapach np. w module Floor Planer'a. Warto podkreślić, że produkt ChipScope jest produktem przeznaczonym tylko dla układów FPGA firmy Xilinx i że jest produktem komercyjnym – płatnym. Ponadto wymusza na konsumencie stosowanie wybranego interface'u (np. kabla MultiLinx), co w niektórych produktach np. płyty XSV firmy Xess ogranicza możliwości projektowe.

Analizator LA_RCS ma jednak szereg zalet w porównaniu z produktem ChipScope. Najważniejsza z nich to zastosowanie kompresji danych RLC, dzięki czemu pozorna pamięć analizatora bardzo często ulega wielokrotnemu zwiększeniu. Ponadto zastosowano zaawansowaną logikę zezwolenia zegara, która umożliwia rejestrowanie tylko wybranych przebiegów. Ponadto zarejestrowane sygnały są wyświetlane w symulatorze języka VHDL, dzięki czemu zarejestrowane sygnały mogą służyć jako wektor wymuszeń lub też mogą być porównywane w prosty sposób z przebiegami otrzymanymi podczas symulacji funkcjonalnej.

6. PODSUMOWANIE

Zaprezentowany system APSI stanowi kompletne narzędzie do projektowania układów cyfrowych nadzorowanych przez komputer PC. Zaproponowano prosty język skryptu służący do komunikacji z platformą sprzętową oraz metodę niezależnej symulacji zarówno platformy sprzętowej jak i programowej. Metoda ta umożliwia proste symulowanie całego systemu jak też poszczególnych jego elementów. Omawiany system jest szeroko wykorzystywany przez Zespół Rekonfigurowalnych Systemów Obliczeniowych w AGH, podczas projektowania własnych modułów sprzętowych i zdecydowana większość wektorów wymuszeń dla celów symulacji jest uzyskiwana dzięki zastosowaniu skryptu APSI i kosymulacji heterogenicznej. Omawiany system dzięki temu zdecydowanie skraca etap projektowy, dla którego proces symulacji jest bardzo często najbardziej czasochłonny.

Jedną z zalet proponowanego systemu jest blokowa budowa platformy sprzętowej z wykorzystaniem środowiska EDK. Dzięki temu integracja różnych projektów odbywa

się w sposób uproszczony. Wiąże się to jednak z dodatkowym wysiłkiem projektanta związanym ze standaryzacją projektu i integracją wykonywanego modułu ze środowiskiem EDK, które jest ciągle na etapie rozwoju i zawiera szereg wad i błędów, które powoli są korygowane.

Bardzo często rzeczywisty projekt nie działa poprawnie pomimo tego, że proces symulacji przebiega poprawnie – z doświadczenia około 30% prostych projektów i około 90% skomplikowanych projektów. W tym wypadku zastosowanie analizatora jest bardzo często jedynym rozwiązaniem. Autorski analizator LA_RCS ma szereg nowatorskich rozwiązań takich jak: kompresja danych RLC, zastosowanie logiki zezwolenia zegara czy też integracja interface'u wyświetlającego dane bezpośrednio z symulatorem VHDL. Niektóre z tych rozwiązań są już powoli adaptowane w rozwiązaniach komercyjnych, np. najnowsza wersja 6.2 ChipScope'a umożliwia zastosowanie zarejestrowanych sygnałów jako wektorów wymuszeń w VHDL.

Wspomniany system, a szczególnie jego platforma sprzętowa, jest ciągle rozwijany i stanowi podstawę do uruchamiania nowych systemów. Bardzo często budowa nowego sytemu wiąże się tylko z budową pojedynczego modułu, pozostałe moduły są dostarczane wraz z systemem APSI, dzięki temu czas projektowania i uruchamiania uległ zdecydowanemu skróceniu. Jest to bardzo ważne ponieważ możliwości współczesnych układów FPGA są w dużej mierze niewykorzystywane z powodu długiego i trudnego procesu projektowego i trudnej do uzyskania integracji całego systemu.
Praca finansowana ze środków Komitetu Badań Naukowych w latach 2003-2005 jako projekt badawczy.

7. BIBLIOGRAFIA

1. Xilinx Co. http://www.xilinx.com : *The Programmable Logic, Data Book.* Xilinx, San Jose, 2003.
2. K.Wiatr: *Akceleracja Obliczeń w Systemach Wizyjnych,* Warszawa, WNT 2003.
3. H. Krupnova, V. Meurou, C. Barchon, C. Serra, F. Morsi: *How Fast Is Rapid FPGA-based Prototyping: Lessons and Challenges from the Digital TV Design Prototypes Project,* Proc. Field-Programmable Logic FPL 2002 Montpellier, France, Sep. 2-4, pp.26-35.
4. M. Gschwind, V. Salapura, D. Maurer: *FPGA Prototyping of a RISC Processor Core for Embedded Applications,* IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 9, no. 2, April 2001, pp. 241-250.
5. R. Goering: *French EDK startup is fluent in co-design* EE Times, Oct. 31, 2003, www.eedesign.com
6. Dolphin Integration, *SUCCESS™ Hardware / Software cosimulation* http://www.dolphin.fr/medal/success/success_overview.html
7. Xess Co. http://www.xess.com/manuals.html
8. OpenCores WISHBONE SoC Interconnection http://www.opencores.org/wishbone/
9. IBM, *CoreConnect™ bus architecture,* http://www-3.ibm.com/chips/products/coreconnect/
10. Xilinx Inc. *ChipScope Pro Software and Cores User Manual,v6.*1 August 29 2003, Xilinx
11. Xilinx Inc. *MicroBlaze Processor Reference Guide, Embedded Development Kit,* EDK 16-th Sep. 2002, Xilinx

E. Jamro, K. Wiatr

ADVANCED PROGRAMMABLE SYSTEMS INTERFACE FOR PROTOTYPING HETEROGENEOUS MODULES WITH FPGA CHIPS

Summary

This paper describes the Advanced Programmable System Interface (APSI), dedicated for FPGA-based boards connected to a PC. The APSI includes: the interpreter for dedicated script language to efficiently communicate with a FPGA-based board; heterogeneous hardware-software co-simulation to simulate either PC or hardware (FPGA-based board) sides; and internal logic state analyzer. The whole APSI system has been design by the authors and significantly seeds up development cycle for the FPGA-based designs. The proposed system contains several novel ideas, e.g. the concept of hardware-software co-simulation, internal logic state analyzer with data compression, clock enable and VHDL-based interface.

Keywords: reconfigurable computing systems, programmable structures, FPGA, digital systems design, hardware-software co-design

KWARTALNIK ELEKTRONIKI I TELEKOMUNIKACJI, 2006, 52, z. 1, ss. 115–128

Implementacja sieci neuronowych w układach programowalnych FPGA dla potrzeb przetwarzania obrazów w czasie rzeczywistym

KAZIMIERZ WIATR [1,2], PAWEŁ CHWIEJ [1]

[1] *Akademia Górniczo-Hutnicza, Katedra Elektroniki, al. Mickiewicza 30, 30-059 Kraków*
[2] *Akademickie Centrum Komputerowe CYFRONET AGH, ul. Nawojki 11, 30-950 Kraków*
email: wiatr@agh.edu.pl, pchwiej@agh.edu.pl

Otrzymano 2005.01.05
Autoryzowano 2005.05.04

Celem niniejszego artykułu jest przedstawienie możliwości implementacji wybranej sieci neuronowej do przetwarzania obrazów w układach programowalnych FPGA. Autorzy zakładają, że uczenie sieci neuronowej następuje w komputerze ogólnego przeznaczenia, natomiast implementacja w FPGA dotyczy sieci neuronowej już nauczonej. Sieć komórkowa łączy w sobie cechy sztucznej sieci neuronowej czyli przetwarzania informacji przy użyciu identycznych elementów i prostej strukturze oraz funkcji z modelem automatów komórkowych, czyli regularną budową i lokalnymi połączeniami międzyelementowymi. Charakterystyczne jest także to, że wagi połączeń są stałe, a sieć wykazuje charakter rekurencyjny. Ponadto sieć taka swoją strukturą dobrze odpowiada architekturze wewnętrznej układów programowalnych FPGA, dzięki czemu wyjątkowo korzystnie przebiega jej implementacja w takich strukturach. W artykule przedstawione zostaną przykładowe implementacje w układach programowalnych FPGA firmy Xilinx. W szczególności zostaną zaprezentowane maksymalne osiągnięte szybkości pracy zaimplementowanych sieci, wnoszone opóźnienie oraz związany z tymi sieciami koszt mierzony wielkością użytych zasobów wewnętrznych układu FPGA i odniesiony do szerokości bitowej słowa wejściowego.

Słowa kluczowe: rekonfigurowalne systemy obliczeniowe, sieci neuronowe, FPGA, układy programowalne, przetwarzanie obrazów

1. WSTĘP

Przetwarzanie obrazów w czasie rzeczywistym wymaga dużych mocy obliczeniowych. Korzystnym jest stosowanie do realizacji algorytmów przetwarzania obrazów dedykowanego sprzętu. Rozwój układów programowalnych pozwala na wyjątkowo dobrą realizacje takich rozwiązań znanych pod nazwą CCM (ang. *Custom Computing Machines*). Szczególnie interesujące jest zastosowane do tego typu obliczeń sieci neuronowych.

Sztuczne sieci neuronowe SSN powstały z interdyscyplinarnej syntezy wyników tradycyjnych obejmujących biologię, fizykę i matematykę. Podstawową cechą różniącą SSN od programów realizujących algorytmiczne przetwarzanie informacji jest zdolność generalizacji czyli uogólniania wiedzy dla nowych danych, nieznanych wcześniej, czyli nie prezentowanych w trakcie nauki. Określa się to także, jako zdolność SSN do aproksymacji wartości funkcji wielu zmiennych w przeciwieństwie do interpolacji możliwej do otrzymania przy przetwarzaniu algorytmicznym. Można to ująć jeszcze inaczej. Np. systemy ekspertowe z reguły wymagają zgromadzenia i bieżącego dostępu do całej wiedzy na temat zagadnień, o których będą rozstrzygały. SSN wymagają natomiast jednorazowego nauczenia, przy czym wykazują one tolerancję na nieciągłości, przypadkowe zaburzenia lub wręcz braki w zbiorze uczącym. Pozwala to na zastosowanie ich tam, gdzie nie da się rozwiązać danego problemu w żaden inny, efektywny sposób.

Przedmiotem prezentowanych badań jest implementacja sieci neuronowej komórkowej w układach programowalnych FPGA, przy czym nie zajmowano się uczeniem tej sieci lecz implementowano sieć już nauczoną. Uczenie sieci w układach FPGA nie jest rozwiązaniem właściwym, ponieważ można to zrobić znacznie efektywniej przy pomocy komputera. Także obliczanie szablonów, do celów jakie ma realizować sieć, powinno odbywać się przy użyciu komputera. W przedstawionych sieciach wykorzystano wartości współczynników jedynie do zbadania możliwości implementacji maksymalnego obszaru jaki może zajmować sieć zdolna do rozpoznawania obrazu. Podano także wyniki otrzymane z implementacji sieci z szablonami przeznaczonymi do uwypuklenia kątów oraz do detekcji krawędzi, dla trzech sieci o różnych wielkościach sygnałów wejściowych i wyjściowych.

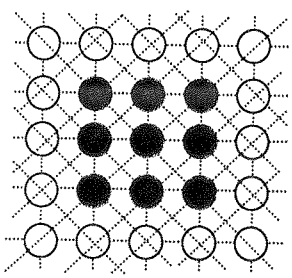
Głównym celem prowadzonych prac było zbadanie możliwości implementacji sieci neuronowej, która pozwala na rozpoznawanie obrazów, w strukturze układu programowalnego FPGA. Wybrana sieć, którą jest sieć neuronowa komórkowa, jest siecią która charakteryzuje się geometryczną budową, podobną, do struktury wewnętrznej układów FPGA. Ważnym faktem jest to, że obecnie, właśnie tego rodzaju sieci są coraz częściej stosowane do przetwarzania obrazów. Analizowane sieci neuronowe komórkowe są sieciami o małych rozmiarach, co ułatwia ich implementację.

2. SIEĆ NEURONOWA KOMÓRKOWA

Rodzajem sieci neuronowych, które znalazły szerokie zastosowanie w różnych dziedzinach jest sieć komórkowa CNN (ang. *Cellular Neural Network*) przedstawiona

przez Yang'a i Chua [9]. Łączy ona cechy sztucznej sieci neuronowej, czyli przetwarzania informacji przy użyciu identycznych elementów o prostej strukturze i funkcji, z modelem automatów komórkowych, czyli regularną budową i lokalnymi połączeniami międzyelementowymi. Charakterystyczne jest także to, że wagi połączeń są stałe a sieć wykazuje charakter rekurencyjny.

Sieć ta jest zbudowana z identycznych elementów połączonych ze sobą w obrębie bliskiego sąsiedztwa, które tworzą regularną architekturę geometryczną. Sygnały w takiej sieci są przetwarzane w sposób równoległy, a oddziaływanie pomiędzy komórkami zachodzi lokalnie. Jednak pomimo lokalnego zasięgu oddziaływań sieć neuronowa komórkowa może realizować przetwarzanie o charakterze globalnym, ponieważ wynik może zależeć od wartości komórek leżących poza obszarem najbliższego sąsiedztwa. Dzieje się tak dlatego, że komórka sterowana jest sygnałem wyjściowym komórek jej sąsiedztwa, które są zmienne w czasie. Przykładową dziedziną zastosowania tego rodzaju sieci jest przetwarzanie obrazów, w tym: segmentacja obrazów rzeczywistych, redukcja szumów, detekcja konturów, detekcja cech i detekcja ruchu.

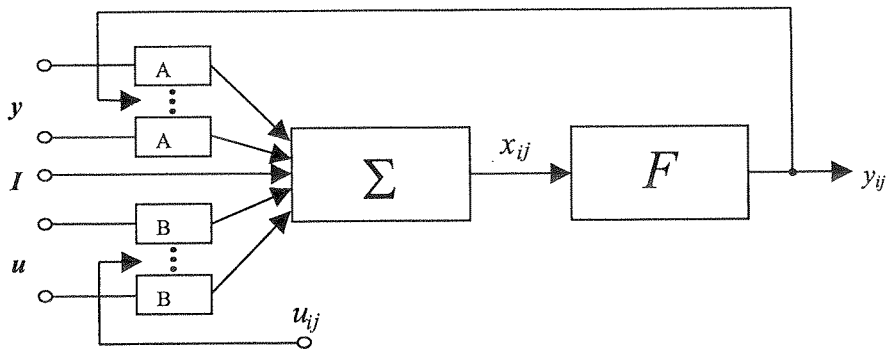


Rys. 1. Fragment sieci neuronowej komórkowej o sąsiedztwie r = 1

Fig. 1. Part of neural network with neighbourhood r = 1

Pojedynczym elementem sieci jest komórka. Komórki tworzą regularną, płaską lub przestrzenną siatkę geometryczną i są połączone ze sobą lokalnymi kanałami (rys. 1).

Informacja jest wprowadzana do wszystkich komórek w sposób równoległy. W sieci zachodzą dynamiczne procesy przejściowe, które spowodowane są wzajemnym oddziaływaniem międzykomórkowym. Po osiągnięciu stanu równowagi przez sieć, osiągamy sygnał wyjściowy, który bardzo często jest sygnałem binarnym. Mimo, że komórki są połączone lokalnie to przetwarzanie ma charakter globalny, bowiem w procesie przetwarzania bierze udział cała sieć. Budowa sieci komórkowej ma charakter regularny, a każda komórka połączona jest z komórką leżącą w jej sąsiedztwie. Wyjątek stanowią komórki brzegowe, które są pobudzane sygnałami wywołanymi w sposób sztuczny. Dobierane są one w zależności od zastosowania danej sieci. Istnieją rozwiązania które łączą komórki z jednego brzegu z komórkami z brzegu przeciwnego tworząc sieć o strukturze toroidalnej.



Rys. 2. Model komórki sieci neuronowej

Fig. 2. Model of neural networks cell

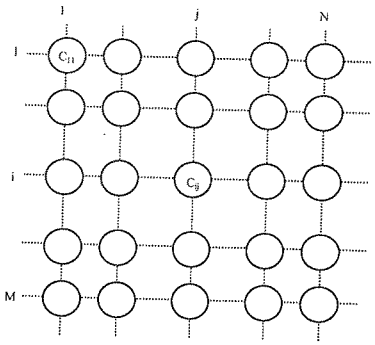
Każda komórka, oprócz sygnałów wejścia i wyjścia, posiada sygnał polaryzacji. Sygnał polaryzacji jest wartością stałą, ale nie jest jednakowy dla każdej komórki w sieci. Funkcja transformująca odwzorowuje stan komórki X_{ij} w stan wyjściowy Y_{ij} (rys. 2). Funkcja ta jest odcinkowo liniowa, obustronnie nasycona, o nachyleniu jednostkowym w otoczeniu początku układu współrzędnych. Dynamikę zmian w czasie sygnału stanu i zmiany sygnału wyjściowego komórki można przedstawić za pomocą równań nieliniowych:

$$C\frac{dx_{ij}(t)}{dt} = -\frac{1}{R}x_{ij}(t) + \sum_{C(k,l)\in N(i,j)} A(i,j;k,l)y_{kj}(t) + \sum_{C(k,l)\in N(i,j)} B(i,j;k,l)u_{kl} + I \qquad (1)$$

$$y_{ij}(t) = \frac{1}{2}(|x_{ij}(t)+1| - |x_{ij}(t)-1|) \qquad (2)$$

gdzie:
i, j – indeksy odnoszące się do komórki będącej obiektem sterowania,
k, l – indeksy odnoszące się do komórek sterujących,
R, C – rezystancja i pojemność określające właściwości dynamiczne sieci.

Każda cela (komórka) jest połączona tylko z sąsiadami. Sąsiednie cele oddziałują na siebie wzajemnie. Intuicyjnie należy znaleźć obszar i oznaczyć go jako N_{ij} (rys. 3). Każda komórka posiada wartość początkową zmiennych stanu x, wejścia u (własne jak i komórek należących do jej sąsiedztwa) i wyjścia y (własne i wyjścia komórek należących do jej sąsiedztwa). Wszystkie te sygnały sterują komórką. Każda z komórek przetwarza sygnały w identyczny sposób. Wynikiem jest ważona suma sygnałów sterujących, która jest poddana nieliniowej transformacji. Stan każdej komórki jest aktywny po czasie $t > 0$, osiągając stan równowagi. Neuronowa sieć komórkowa zwraca zawsze jeden stabilny stan.



Rys. 3. Topologia sieci neuronowej

Fig. 3. Neural networks topology

Wartości wag oraz połączenia komórek są jednakowe dla każdej komórki w sieci, dlatego dla dowolnych dwóch komórek C_{ij} i C_{IJ} współczynniki wagowe są równe. Powoduje to, że można pominąć indeksy ij i IJ i pozostawić jedynie indeksy lk.

3. IMPLEMENTACJA POJEDYNCZEJ KOMÓRKI CNN W UKŁADZIE FPGA

Implementacje w układach FPGA rozpoczęto od zaimplementowania pojedynczej komórki sieci CNN w układach: XC4002XL, XCS05XL i XCV1000 firmy Xilinx. Poddano analizie takie czynniki jak: zużycie zasobów układu i szybkość ze względu na budowę komórki. Najprostszy model komórki zbudowany jest z jednobitowego wejścia i wyjścia, (taki rodzaj komórki nadaje się do budowy sieci wykorzystywanej do analizy pisma, gdzie występują tylko dwa elementy: obiekt = 1, i tło = 0) oraz z czterobitowych współczynników ze znakiem. Współczynniki o takim rozmiarze pozwolą zastosować szablon np. do detekcji krawędzi.

Tabela 1

Wielkość zasobów układu FPGA użytych do implementacji pojedynczej komórki CNN
Size of capacity FPGA structure used for implementation CNN single cell

Elementy układu FPGA	4 bit. we/wy. 4 bit. współ. XCS40XL		8 bit. we/wy. 8 bit. współ. XCS40XL		4 bit. we/wy. 4 bit. współ. XC4085XL		1 bit. we/wy. 4 bit. współ. XCS05XL	
	szyb.	obsz.	szyb.	obsz.	szyb.	obsz.	szyb.	obsz.
CLBs	114	114	116	116	114	114	39	39
Przerzutniki	2	2	2	2	2	2	2	2
Zatrzaski	5	5	9	9	5	5	5	5
IOBs	57	57	101	101	57	57	17	17
4 we LUT	191	191	195	195	191	191	59	59
Liczba bramek	2513	2513	2557	2557	2513	2513	811	811

Rozbudowując komórkę zwiększono rozmiar wejścia i wyjścia do 4 bitów ze znakiem, przy współczynnikach 4-bitowych ze znakiem, a następnie wejście i wyjście zwiększono do 8 bitów ze znakiem, przy 8-bitowych współczynnikach ze znakiem. Zastosowano funkcję aktywacji signum. W tab. 1 i 2 znajdują się wyniki otrzymane w trakcie implementacji. W kolumnie „szyb." zawarte są wyniki uzyskane podczas optymalizacji ze względu na szybkość, a w kolumnie „obsz." wyniki dla optymalizacji ze względu na obszar.

Tabela 2

Szybkość pracy pojedynczej komórki CNN zaimplementowanej w układzie FPGA
Speed of CNN single cell implemented in FPGA structure

Parametry czasowe	4 bit. we/wy. 4 bit. współ. XCS40XL		8 bit. we/wy. 8 bit. współ. XCS40XL		4 bit. we/wy. 4 bit. współ. XC4085XL		1 bit. we/wy. 4 bit. współ. XCS05XL	
	szyb.	obsz.	szyb.	obsz.	szyb.	obsz.	szyb.	obsz.
Max częstotliwość [MHz]	25,172	25,172	25,158	25,158	11,002	11,002	51,962	51,962
Max opóźnienie [ns]	8,293	8,293	8,360	8,360	26,493	26,493	3,382	3,382

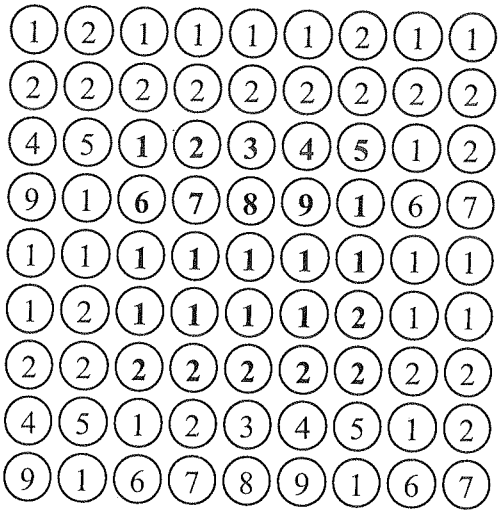
4. IMPLEMENTACJA NEURONOWEJ SIECI KOMÓRKI W UKŁADZIE FPGA

Zaimplementowanie neuronowej sieci komórkowej CNN jest możliwe jedynie w układach serii Virtex, ponieważ poprzednie serie posiadają zbyt małą wielkość zasobów logicznych i połączeniowych, aby mogły pomieścić strukturę sieci. Analizę przeprowadzono dla układu XCV1000, implementując dwie sieci o rozmiarach sąsiedztwa r = 1 i r = 2 w różnych konfiguracjach:

a) sieć r = 1, wejścia i wyjścia 1-bitowe, 4-bitowe współczynniki ze znakiem,
b) sieć r = 1, wejścia i wyjścia 4-bitowe ze znakiem, 4-bitowe współczynniki ze znakiem,
c) sieć r = 1, wejścia i wyjścia 8-bitowe ze znakiem, 8-bitowe współczynniki ze znakiem,
d) sieć r = 2, wejścia i wyjścia 1-bitowe, 4-bitowe współczynniki ze znakiem,
e) sieć r = 2, wejścia i wyjścia 8-bitowe, 8-bitowe współczynniki ze znakiem.

W każdej sieci zastosowano funkcje aktywacji signum. Sieci posiadają strukturę toroidalną. Przykład rozmieszczenia komórek w takiej sieci, przedstawiony jest na rysunku 4. Centralny element sieci stanowi komórka 13, sieć posiada 25 komórek wzajemnie odziaływujących na siebie.

W tab. 3 i 4 przedstawiono wyniki implementacji sieci w układzie XCV1000. Sieć o rozmiarze r = 2, wejściu i wyjściu 8-bitowym i współczynnikach 8-bitowych nie mieści się w zasobach układu XCV1000 i dlatego w tabelach nie zamieszczono wyników, związanych z jej implementacją.



Rys. 4. Model sieci komórkowej o promieniu r = 2 i o strukturze toroidalnej

Fig. 4. Model of neural networks with radius r = 2 and toroidal structure

Tabela 3

Wielkość zasobów układu FPGA użytych do implementacji sieci CNN
Size of capacity FPGA structure used for implementation CNN network

Elementy układu FPGA	sieć r = 1, 1 bit. we/wy. 4 bit. współ. XCV1000		sieć r = 1, 4 bit. we/wy. 4 bit. współ. XCV1000		sieć r = 1, 8 bit. we/wy. 8 bit. współ. XCV1000		sieć r = 2, 1 bit. we/wy. 4 bit. współ. XCV1000	
	szyb.	obsz.	szyb.	obsz.	szyb.	obsz.	szyb.	obsz.
Slices	746	742	1318	1317	2863	2862	9511	9511
Przerzutniki	18	18	36	36	81	81	150	150
Zatrzaski	9	9	27	27	72	72	125	125
IOBs	64	64	1010	100	217	217	176	176
4 we LUT	516	508	2173	2173	4459	4459	9565	9565
Liczba bramek	9117	9069	25530	25530	53574	53574	146827	146827

Sieć neuronowa komórkowa oprócz połączeń międzykomórkowych oraz sygnałów wejścia/wyjścia, posiada także układy i połączenia sterujące, sprawdzające czy wszystkie komórki zakończyły obliczenia i czy można wysłać wynik na wyjścia komórek. Posiada także możliwość zerowania swoich wartości Na rysunku 4 nie umieszczono połączeń, a także sterowania komórkami, ponieważ liczba połączeń zniekształciłaby obraz sieci.

Poniżej przedstawiono wyniki otrzymane po zaimplementowaniu sieci o rozmiarze 3 x 3, które realizują określone funkcje związane z przetwarzaniem obrazu.

Tabela 4

Szybkość pracy sieci CNN zaimplementowanej w układzie FPGA
Speed of CNN network implemented in FPGA structure

Parametry czasowe	sieć r = 1, 1 bit. we/wy. 4 bit. współ. XCV1000		sieć r = 1, 4 bit. we/wy. 4 bit. współ. XCV1000		sieć r = 1, 8 bit. we/wy. 8 bit. współ. XCV1000		sieć r = 2, 1 bit. we/wy. 4 bit. współ. XCV1000	
	szyb.	obsz.	szyb.	obsz.	szyb.	obsz.	szyb.	obsz.
Max częstotliwość [MHz]	16,762	16,437	14,298	13,854	11,887	11,830	5,274	5,274
Max opóźnienie [ns]	11,421	14,617	15,276	15,113	16,016	10,257	19,963	19,963

Zaimplementowano sieć dla wybranego szablonu (3.), który służy do uwypuklenia kątów. Porównuje ona trzy sieci o tym samym szablonie:

1. sieć 2-bitowe wejście i wyjście ze znakiem,
2. sieć 3-bitowe wejście i wyjście ze znakiem,
3. sieć 8-bitowe wejście i wyjście ze znakiem.

Szablon:

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 4 & -1 \\ -1 & -1 & -1 \end{bmatrix}, \mathrm{I} = -5 \tag{3}$$

wymaga użycia 3-bitowych współczynników, ale tylko 11 z 19 jest niezerowych.

Otrzymane wyniki przedstawione są w tab. 5 i 6. Tab. 5 przedstawia porównanie zajmowanego obszaru układu przez daną sieć. Tab. 6 przedstawia porównanie szybkości działania sieci.

Tabela 5

Zasoby FPGA użyte do implementacji sieci neuronowej z szablonem do uwypuklenia kątów
Size of capacity FPGA structure used for implementation NN with pattern for enhance corner

Elementy układu FPGA	2 bitowe we/wy ze znakiem	3 bitowe we/wy ze znakiem	8 bitowe we/wy ze znakiem
Slices	514	547	600
Przerzutniki	43	45	90
Zatrzaski	34	36	81
IOBs	55	109	244
4 we LUT	812	885	974
Liczba bramek	9760	10245	11382

Drugim rodzajem sieci, jest sieć z szablonem (4.), służąca do detekcji krawędzi. Tak jak w poprzednim przypadku liczba niezerowych elementów wynosi 11.

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 7 & 1 \\ 1 & 1 & 1 \end{bmatrix}, B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \mathrm{I} = 5 \tag{4}$$

Tabela 6

Szybkość pracy sieci neuronowej z szablonem do uwypuklenia kątów zaimplementowanej w FPGA
Speed of NN with pattern for enhance corner implemented in FPGA

Parametry czasowe	2 bitowe we/wy ze znakiem	3 bitowe we/wy ze znakiem	8 bitowe we/wy ze znakiem
Max częstotliwość [MHz]	22,702	22,362	22,142
Max opóźnienie [ns]	12,351	9,385	8,621

Dokonano implementacji trzech sieci tak jak w poprzednim przykładzie. Otrzymane wyniki przedstawiono w tab. 7 i 8. Tab. 7 przedstawia porównanie zajmowanego obszaru układu przez daną sieć. Tab. 8 przedstawia porównanie szybkości działania sieci.

Tabela 7

Zasoby układu FPGA użyte do implementacji sieci neuronowej z szablonem do detekcji krawędzi
Size of capacity FPGA structure used for implementation NN with pattern for edge detect

Elementy układu FPGA	2 bitowe we/wy ze znakiem	3 bitowe we/wy ze znakiem	8 bitowe we/wy ze znakiem
Slices	489	626	733
Przerzutniki	27	45	90
Zatrzaski	18	36	81
IOBs	53	105	211
4 we LUT	769	931	1156
Liczba bramek	9294	11175	13542

Tabela 8

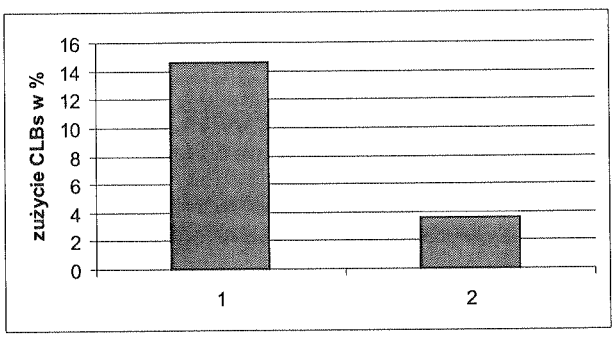
Szybkość pracy sieci neuronowej z szablonem do detekcji krawędzi zaimplementowanej w FPGA
Speed NN with pattern for edge detect implemented in FPGA

Parametry czasowe	2 bitowe we/wy ze znakiem	3 bitowe we/wy ze znakiem	8 bitowe we/wy ze znakiem
Max częstotliwość [MHz]	25,401	23,910	21,797
Max opóźnienie [ns]	10,542	11,829	8,757

5. ANALIZA WYNIKÓW

Przeprowadzone badania nad możliwością implementacji sieci neuronowych komórkowych w układach FPGA, pozwoliły na porównanie układów FPGA pod względem ich przydatności to tego celu. Okazało się, że do implementacji sieci tylko układy serii Virtex spełniają warunki wystarczającej wielkości zasobów, pozwalające na zaimplementowanie całej sieci. Na rysunku 5 przedstawiono obszar jaki zajmuje jedna komórka z 4-bitowym wejściem i wyjściem oraz 4-bitowymi współczynnikami

(1 – układ XCS40XL Spartan, 2 – układ XC4085XL). Wyniki pozwalają sformułować wniosek, że w układzie Spartan pojedyncza komórka zajmuje procentowo znaczny obszar, co eliminuje ten rodzaj układów do implementacji sieci. Zaznaczyć należy, że w każdym układzie taka sama sieć zajmuje dokładnie tyle samo powierzchni. Atutem układu Spartan jest to, że znacznie przewyższa on układ serii XC4000 szybkością, ponieważ maksymalna szybkość dla tego układu przy implementacji powyższej komórki wynosiła 25,172 MHz, a dla konkurenta 11,002 MHz.



Rys. 5. Obszar układu FPGA zajmowany przez pojedynczą komórkę CNN

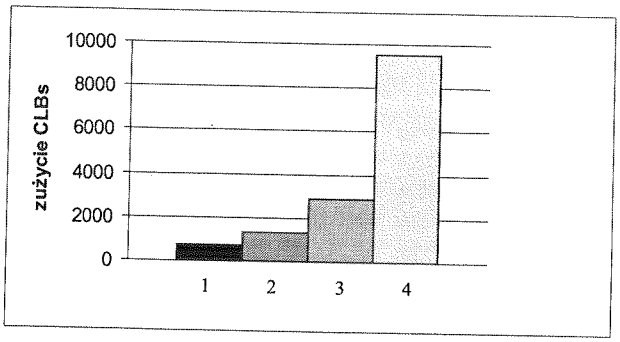
Fig. 5. FPGA area used for single CNN cell

Analizie poddano również sieci komórkowe w różnych konfiguracjach zaimplementowane w układzie Virtex XCV1000. Zużycie obszaru przez poszczególne sieci przedstawiono na rysunku 6, a procentowe zajęcie obszaru układu na rysunku 7,

gdzie:

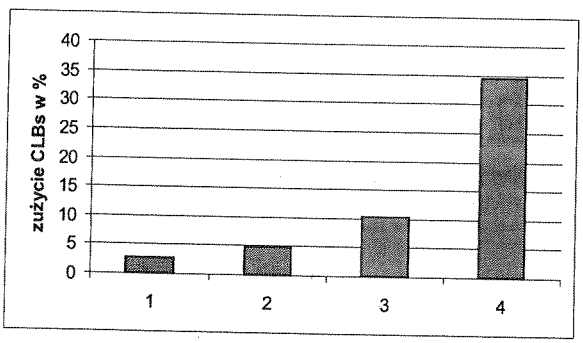
1 – sieć r = 1, wejścia i wyjścia 1-bitowe, 4-bitowe współczynniki ze znakiem,
2 – sieć r = 1, wejścia i wyjścia 4-bitowe ze znakiem, 4-bitowe współczynniki ze znakiem,
3 – sieć r = 1, wejścia i wyjścia 8-bitowe ze znakiem, 8-bitowe współczynniki ze znakiem,
4 – sieć r = 2, wejścia i wyjścia 1-bitowe, 4-bitowe współczynniki ze znakiem.

Zaimplementowanie sieci o promieniu r = 2 pochłania znaczny obszar. Implementacja tej sieci o 8-bitowych wejściach i wyjściach oraz 8-bitowych współczynnikach nie powiodła się, ponieważ sieć przekroczyła rozmiary układu. Rysunek 8 przedstawia maksymalne szybkości z jakimi może pracować układ zawierający daną sieć. Seria pierwsza przedstawia implementacje przed optymalizacją ze względu na szybkość, druga po optymalizacji. Na rysunku 9 umieszczone są przyrosty prędkości po optymalizacji ze względu na szybkość. Optymalizacja rozbudowanej sieci o promieniu r = 2 nie przyniosła efektu. Spowodowane jest to znacznym zajęciem obszaru przez sieć i ograniczenie zmian dla układu podczas optymalizacji. W tab. 9 przedstawione są wartości przyrostu prędkości.



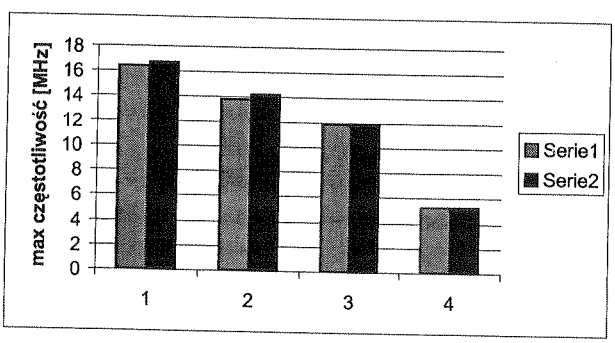
Rys. 6. Obszar FPGA zajmowany przez CNN

Fig. 6. FPGA area used for CNN network



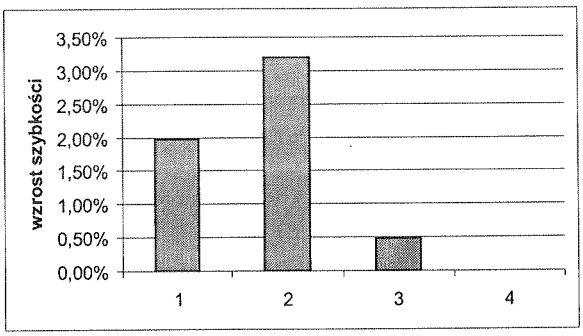
Rys. 7. Wykorzystanie FPGA przez CNN

Fig. 7. FPGA utilize for CNN



Rys. 8. Szybkości pracy sieci CNN w układzie XCV1000

Fig. 8. Speed CNN network implemented in XCV1000



Rys. 9. Przyrosty szybkości pracy sieci CNN po optymalizacji

Fig. 9. CNN network speed increment after optimization

Powyższe sieci, są sieciami skonfigurowanymi na maksymalne wartości, ponieważ w rzeczywistości, nie wszystkie współczynniki szablonu są niezerowe, a także nie jest konieczne zapisywanie współczynników np. na 8 bitach, co zajmuje znaczny obszar i pogarsza prędkość układu. Przedstawione w analizie dwa rodzaje sieci: do uwypuklania kątów i detekcji krawędzi, o 11 niezerowych współczynnikach szablonu, charakteryzują się dobrymi, w porównaniu z innymi sieciami szybkościami, przy małym zużyciu obszaru układu. Szybkości, nawet przy 8-bitowym wejściu i wyjściu, nie są mniejsze niż 20 MHz. Dla sieci przeznaczonej do uwypuklania kątów szybkość tylko nieznacznie zmniejsza się przy wzroście liczby bitów. Zastosowanie komórkowego procesora obrazu z wykorzystaniem właśnie sieci CNN daje realną szansę na jego wykorzystanie do przetwarzania obrazów w czasie rzeczywistym.

Tabela 9

Przyrost prędkości pracy sieci CNN po optymalizacji
CNN network speed increment after optimization

	1	2	3	4
Wzrost szybkości	1,98%	3,20%	0,48%	0%

6. PODSUMOWANIE

Podsumowując wyniki przedstawionych prac należy stwierdzić, że sieć neuronowa komórkowa CNN bardzo dobrze nadaje się do implementacji w układach FPGA. Budowa zbyt dużej sieci neuronowej i zaimplementowanie jej w układzie FPGA, choć jest to możliwe wykorzystując układ XC2V8000, nie jest dobrym rozwiązaniem, ponieważ obniża to znacznie prędkość układu. Rozwiązaniem lepszym wydaje się stosowanie komórkowego procesora obrazu, wykorzystując sekwencyjny charakter jego pracy. Należy także zastosować odpowiedni interfejs umożliwiający przetworzenie obrazu do

postaci cyfrowej. Wstępne przetwarzanie może dokonywać się już w sieci komórkowej. Zaimplementowana w układzie FPGA sieć neuronowa może służyć do różnych zadań związanych z przetwarzaniem obrazu, należy tylko przeprogramowywać szablony odpowiednio dla danego zadania.

Dalszy rozwój przedstawionych prac to jeszcze większe połączenie procesora komórkowego z własnościami sieci i z dalszym rozwojem układów FPGA. Przy obecnym bardzo dynamicznym rozwoju układów programowalnych można sądzić, że jest możliwe zaimplementowanie w tego rodzaju układach, nie tylko sieci neuronowej komórkowej, ale także układy wspomagające tą sieć: dodatkowe obszary pamięci przechowujące szablony, układy wstępnego przetwarzania i inne. Praca finansowana ze środków Komitetu Badań Naukowych w latach 2003–2005 jako projekt badawczy.

7. BIBLIOGRAFIA

1. T. Kacprzak, K. Ślot: *Sieci neuronowe komórkowe – Teoria, projektowanie, zastosowania,* Warszawa-Łódź, PWN 1995.
2. K. Ślot: *Sieci neuronowe komórkowe: efektywne narzędzia przetwarzania informacji obrazowej,* Wydawnictwo Politechniki Łódzkiej 1999, Zeszyty Naukowe nr 819.
3. www.isi.ee.ethz.ch/~haenggi/CNN_web/CNNsim_adv.html
4. L. O. Chua, C. W. Wu: *On the Universe of Stable Cellular Neural Networks,* International Journal of Circuit Theory and Applications, vol.20 September – October 1992, pp. 497-518.
5. K. Ślot: *Analiza projektowania i synteza jednowarstwowych komórkowych sieci neuropodobnych dla celów przetwarzania obrazów,* Rozprawa doktorska, Politechnika Łódzka, Łódź 1993.
6. M. Ogorzałek, A. Dąbrowski: *Theoretical and Experimental Studies of Oscillations in simple CNN Structures,* Proceedings of Second Int. Workshop on Cellular Neural Networks and their Applications CNNA'92, Munich, Germany, October 14–16, 1992, pp. 123-128.
7. T. Roska, J. Vandewalle: *Cellular Neural Networks,* Chichester: J. Wiley & Sons 1993.
8. F. A. Savaci, J. Vandewalle: *On the Stability Analysis of Cellular Neural Networks,* IEEE Trans. Circuits and System I, vol. 40 , March 1993, pp. 213-214.
9. L. O. Chua, L. Yang: *Cellular Neural Networks: Theory and Applications,* 1988.
10. P. P. Civalleri, M. Gilli, L. Pandolf: *On Stability of Cellular Neural Networks with Delay,* 1993.
11. T. Boros, K. Lotz, A. Radvanyi, T. Roska: *Some useful, New, Nonlinear and Delay Type Templates,* Report DNS-1991, Dual and Neural Computing System Laboratory, Hungary Academy of Sciences, Budapest, Hungry, 1991.
12. S. Jankowski, R. Wańczuk: *Nonlinear CNN Cloning Template for Image Thicking,* Proceedings of Second Int. Workshop on Cellular Neural Networks and their Applications CNNA'92, Munich, Germany 1992.
13. T. Roska, L. O. Chua: *Cellular Neural Networks with Nonlinear and Delay-Type Template Elements,* Proceedings of Int. Workshop on Cellular Neural Networks and their Applications CNNA'90, Budapest, Hungary 1990.
14. L. O. Chua, B. Shi: *Multiple Layer Cellular Neural Networks,* A Tutorial, Memo No.UCB/ERL M90/113, University of California, Berkley, 1990.
15. L. O. Chua: CNN: *A Vision of Complexity,* Int. Journ. Of Bifurcation and Chaos, 1997.
16. T. Roska, L. O. Chua: *The CNN Universal Machine: An analogic array computer,* IEEE, 1993.
17. K. Wiatr: *Sprzętowe implementacje algorytmów przetwarzania obrazów w systemach wizyjnych czasu rzeczywistego,* Kraków, Wyd. Naukowo-Dydaktyczne AGH 2002.

18. www.xilinx.com/partinfo/databook.htm
19. K. W i a t r: *Akceleracja obliczeń w systemach wizyjnych,* Warszawa, WNT 2003.

K. WIATR, P. CHWIEJ

NEURAL NETWORKS IMPLEMENTATION IN FPGA PROGRAMMABLE CHIPS FOR REAL-TIME IMAGE PROCESSING

Summary

In this paper the implementation of fragment digital Cellular Neural Network (CNN) for image processing on the Field Programmable Gate Array (FPGA) and it's experiment results are present. The high processing speed of the network is use to provide real time processing. Results shows that the architecture CNN and FPGA and implementation has good corespondent. The above presented networks are configured in maximum values because, in reality, not all the coefficients of the pattern are non-zero. It is also unnecessary to record the coefficients in 8 bits. This solution occupies considerable area and decreases the system speed. In the analysis, We have introduced two kinds of network: for angle embossment and edge detection. They have 11 non- zero pattern coefficients and they characterize, in comparison to the preceding networks, in good speeds, at little waste of the system area. The speeds, even at 8 bit input and output do not fall below 20 MHz. For the angle embossment network the speed only slightly decreases during the bit increase. The use of the cellular image processor with the application of these networks gives a real chance for the physical utilization of the network. Summing up the results of our work, we can assert that cellular neural networks are suitable for the implementation in FPGA systems. However, the utilization of a network implemented in FPGA systems has to take place with cooperation with other systems. The construction of too large a neural network and its implementation in FPGA system, despite the possibility of using XC2V8000, is not a good solution because it considerably decreases the system speed.

Keywords: reconfigurable computing systems, neural networks, FPGA, programmable chips, image processing

INFORMACJE DLA AUTORÓW

Redakcja przyjmuje do publikowania prace oryginalne, przeglądowe i monograficzne wchodzące w zakres szeroko pojętej elektroniki. Ponieważ KWARTALNIK ELEKTRONIKI I TELEKOMUNIKACJI jest czasopismem Komitetu Elektroniki i Telekomunikacji Polskiej Akademii Nauk, w związku z tym na jego łamach znajdują się prace naukowe dotyczące podstaw teoretycznych i zastosowań z zakresu elektroniki, telekomunikacji, mikroelektroniki, optoelektroniki, radiotechniki i elektroniki medycznej.
Artykuły powinno charakteryzować oryginalne ujęcie zagadnienia, własna klasyfikacja, krytyczna ocena (teorii lub metod), omówienie aktualnego stanu, lub postępu danej gałęzi techniki oraz omówienie perspektyw rozwojowych. Artykuły publikowane w innych czasopismach nie mogą być kierowane do druku w Kwartalniku Elektroniki i Telekomunikacji w drugiej kolejności zgłoszenia.
Objętość artykułu nie powinna przekraczać 30 stron po około 1800 znaków na stronie, w tym rysunki i tabele.

Wymagania podstawowe.
Artykuły należy nadsyłać na wyraźnym, jednostronnym, czarno-białym wydruku komputerowym. Wydruk w formacie A4 powinien mieć znormalizowaną liczbę wierszy i znaków w wierszu (30 wierszy po 60 znaków w wierszu), w dwóch egzemplarzach, w języku polskim lub angielskim wybranym przez autora. Do wydruku powinna być dołączona dyskietka z elektronicznym tekstem artykułu. Preferowane edytory to WORD 6 lub 8.
Układ artykułu (w wersji podstawowej) musi być następujący:
- Tytuł.
- Autor (imię i nazwisko autora/ów).
- Miejsce pracy (nazwa instytucji, miejscowość, adres. + ew. adres elektroniczny (e-mail).
- Zwięzłe streszczenie powinno być w języku takim, w jakim jest pisany artykuł (wraz ze słowami kluczowymi).
- Tekst podstawowy powinien mieć następujący układ:
  1. WPROWADZENIE
  2. np. TEORIA
  3. np. WYNIKI NUMERYCZNE
  3.1. .....
  3.2. .....
  4. .....
  5. .....
  6. PODSUMOWANIE
  7. ew. PODZIĘKOWANIA
  8. BIBLIIOGRAFIA
- Układ streszczenia w dodatkowej wersji językowej powinien być następujący:
  AUTOR (inicjał imienia i nazwisko).
  TYTUŁ (w języku angielskim – o ile artykuł pisany jest w języku polskim i na odrwót).
Obszerne do 3600 znaków streszczenia (wraz z słowami kluczowymi) w języku:
  a. angielskim, gdy artykuł pisany jest w języku polskim.
  b. polskim, gdy artykuł pisany jest w języku angielskim.
Streszczenie to powinno pozwolić czytelnikowi na uzyskanie istotnych informacji zawartych w pracy. Z tego względu w streszczeniu tym mogą być cytowane numery istotnych wzorów, rysunków i tabel zawartych w podstawowej wersji językowej.
- Wszystkie strony muszą mieć numerację ciągłą.

Sposób pisania tekstu.
Tekst powinien być pisany bez używania wyróżnień, a w szczególności nie dopuszcza się spacjowania, podkreślania i pisania tekstu dużymi literami z wyjątkiem wyrazów, które umownie pisze się dużymi literami (np. FORTRAN). Proponowane wyróżnienia Autor może zaznaczyć w maszynopisie zwykłym ołówkiem za pomocą przyjętych znaków adjustacyjnych, np. podkreślenie linią przerywaną oznacza spacjowanie (rozstrzelenie), podkreślenie linią ciągłą – pogrubienie, podkreślenie wężykiem — kursywa.

Tekst powinien być napisany z podwójnym odstępem między wierszami, tytuły i podtytuły małymi literami. Marginesy z każdej strony powinny mieć około 35 mm. Wielkość czcionki wydruku powinna być zbliżona co najmniej co wielkości czcionki maszyny do pisania (minimum 12 punktów). Przy podziale pracy na rozdziały i podrozdziały cyfrowe ich oznaczenia nie powinny być większe niż II stopnia (np. 4.1.1.).

Sposób pisania tabel.
Tabele powinny być pisane na oddzielnych stronach. Tytuły rubryk pionowych i poziomych powinny być napisane małymi literami z podwójnym odstępem między wierszami. Przypisy (notki) dotyczące tabel należy pisać bezpośrednio pod tabelami. Tabele należy numerować kolejno liczbami arabskimi, u góry każdej tabeli podać tytuł dwujęzyczny. W pierwszej kolejności w podstawowej wersji językowej, a później w dodatkowej wersji językowej. Tabele umieścić na końcu maszynopisu. Przyjmowane są tabele algorytmów i programy na wydrukach komputerowych. W tym przypadku zachowany jest ich oryginalny układ. Tabele powinny być cytowane w tekście.

Sposób pisania wzorów matematycznych.
Rozmieszczenie znaków, cyfr, liter i odstępów powinno być zbliżone do rozmieszczenia elementów druku. Wskaźniki i wykładniki potęg powinny być napisane wyraźnie i być prawidłowo obniżone lub podwyższone w stosunku do linii wiersza podstawowego. Znaki nad literami i cyframi, całkami i in. symbolami (strzałki, linie, kropki, daszki) powinny być umieszczone dokładnie nad tymi elementami, do których się odnoszą. Numery wzorów cyframi arabskimi powinny być kolejne i umieszczone w nawiasach okrągłych z prawej strony. Nazwy jednostek, symbole literowe i graficzne powinny być zgodne z wytycznymi IEE (International Electronical Commision) oraz ISO (International Organization of Standarization).

Powołania.
Powołania na publikacje powinny być umieszczone na ostatnich stronach tekstu pod tytułem „Bibliografia", opatrzone numeracją kolejną bez nawiasów. Numeracja ta powinna być zgodna z odnośnikami w tekście artykułu.
Przykłady opisu publikacji:

- periodycznej — 1. F. Valdoni: A new milimetre wave satelite. E.T.T. 1990, vol. 2, no 5, pp. 141–148
- nieperiodyczne — 2. K. Andersen: A resource allocation framework. XVI International Symposium, Stockholm (Sweden), may 1991, paper A 2,4
- książki — 3. Y.P. Tvidis: Operation and modeling of the MOS transistors. New York, McGraw-Hill, 1987, p. 553

Materiały ilustracyjne.
Rysunki powinny być wykonane wyraźnie, na papierze gładkim lub milimetrowym w formacie nie mniejszym niż 9 × 12 cm. Mogą być także w postaci wydruku komputerowego (preferowany edytor Corel Draw). Fotografie lub diapozytywy przyjmowane są raczej czarno-białe w formacie nie przekraczającym 10 × 15 cm. Na marginesie każdego rysunku i na odwrocie fotografii powinno być napisane ołówkiem imię i nazwisko Autora oraz skrót tytułu artykułu, do którego są przeznaczone oraz numer rysunku. Spis podpisów pod rysunki i fotografie powinny być umieszczone na oddzielnej stronie. Podpisy pod rysunkami (fotografiami) powinny być dwujęzyczne: w pierwszej kolejności w podstawowej wersji językowej, a później w dodatkowej wersji językowej. Rysunki powinny być cytowane w tekście.

Uwagi końcowe.
Na odrębnej stronie powinny być podane następujące informacje:
- adres do korespondencji z kodem pocztowym (domowy lub do miejsca pracy),
- telefon domowy i/lub do miejsca pracy,
- adres e-mailowy (jeśli autor posiada).

Autorowi przysługuje bezpłatnie 20 odbitek artykułu. Dodatkowe egzemplarze odbitek, lub cały zeszyt Autor może zamówić u wydawcy na własny koszt.

Autora obowiązują korekta autorska, którą powinien wykonać w ciągu 3 dni od daty otrzymania tekstu z Redakcji oraz zwrócić osobiście lub listownie pod adres Redakcji. Korekta powinna być naniesiona na przekazanych Autorowi szpaltach na marginesach ew. na osobnym arkuszu w przypadku uzupełnień tekstu większych niż dwa wiersze. W przypadku nie zwrócenia korekty w terminie, korektę przeprowadza Redakcja Techniczna Wydawcy. Redakcja prosi Autorów o powiadomienie ją o zmianie miejsca pracy i adresu prywatnego.

INFORMATION FOR AUTHORS OF K.E.T.

The editorial stuff will accept for publishing only original monographic and survey papers concerning widely understood electronics. Because of the fact that KWARTALNIK ELEKTRONIKI I TELEKOMUNIKACJI is a journal of the Committee for Electronics and Telecommunications of Polish Academy of Scince, it presents scientific works concerning theoretical bases and applications from the field of electronics, telecommunications, microelectronics, optoelectronics, radioelectronics and medical electronics.
Articles should be characterised by original depiction of a problem, its own classification, critical opinion (concerning theories or methods), discussion of an actual state or a progress of a given branch of a technique and discussion of development perspectives.
An article published in other magazines can not be submitted for publishing in K.E.T. TThe size of an article can not exceed 30 pages, 1800 character each, including figures and tables.

Basic requirements

The article should be submitted to the editorial stuff as a one side, clear, black and white computer printout in two copies. The article should be prepared in English or Polish. Floppy disc with an electronic version of the article should be enclosed. Preferred wordprocessors: WORD 6 or 8.
Layout of the article.

- Title.
- Author (first name and surname of author/authors).
- Workplace (institution, adress and e-mail).
- Concise summary in a language article is prepared in (with keywords).
- Main text with following layout:
  - Introduction
  - Theory (if applicable)
  - Numerical results (if applicable)
  - Paragraph 1
  - Paragraph 2
  - .....
  - .....
  - Conclusions
  - Acknowledgements (if applicable)
  - References
- Summary in additional language:
  - Author (firs name initials and surname)
  - Title (in Polish, if article was prepared in English and vice versa)
  - Extenstive summary, hawever not exceeching 3600 characters (along with keywords) in Polish, if artide wos prepared in English and vice versa). The summary should be prepared in a way allowing a reader to obtaoin essential information contained in the artide. For that reason in the summary author can place numbers of essential formulas, figures and tables from the article.

Pages should have continues numbering.

Main text

Main text can not contain formatting such as spacing, underlining, words written in capital letters (except words that are commonly written in capital letters). Author can mark suggested formatting with pencil on the margin of the article using commonly accepted adjusting marks.
Text should be written with double line spacing with 35 mro left and right margin. Titles and subtitles should be written with small letters. Titles and subtitles should be numberd using no mor than 3 levels (i.e. 4.1.1.).

Tables

Tables with their titles should be places on separate page at the end of the article. Titles of rows and columns should be written in small letters with double line spacing. Annotations concerning tables