

P3057/94



ISSN 0239-8044

**1**  
**1994**

**Techniki  
Komputerowe**  
**BIULETYN INFORMACYJNY**



**INSTYTUT MASZYN MATEMATYCZNYCH  
WARSZAWA 1994**



# Techniki Komputerowe

## BIULETYN INFORMACYJNY



P3057 / 94

Rok XXIX, Nr 1, 1994

INSTYTUT MASZYN MATEMATYCZNYCH  
WARSZAWA 1994

Wydaje:

INSTYTUT MASZYN MATEMATYCZNYCH

UL. KRZYWICKIEGO 34

02-798 WARSZAWA

TEL. 621.84.41, TLX 81.78.80, FAX 29.92.70

E-MAIL [imasmat@frodo.nask.org.pl](mailto:imasmat@frodo.nask.org.pl)

Copyright © by Instytut Maszyn Matematycznych, Warszawa 1994

Wydanie publikacji dofinansowane  
przez  
Komitet Badań Naukowych

ZAKŁAD POLIGRAFICZNY.

Z. Warchal

Raszyn, ul. Orzechowa 13a

PWA45/98

## Spis treści

	Str.
Ocena widoczności w powietrzu zamglonym metodą pomiaru transmisji światła monochromatycznego, Romuald Synak .....	5
Programowanie zegara w MS-Windows, Marek Kotowski .....	29
Komputerowa sieć sterująca LON, Rafał Obuchowski .....	45
Transputery i OCCAM, Bożena Przyborowska .....	49
Emuloanalizator EMA-51 i jego wykorzystanie do uruchamiania systemów mikroprocesorowych, Tadeusz Sinkiewicz, Andrzej Kojemski, Lech Naumowski .....	69
Rynek informatyczny, a zwłaszcza komputerów osobistych na świecie i w Polsce, Jan Ryżko .....	85





ROMUALD SYNAK

INSTYTUT MASZYN MATEMATYCZNYCH WARSZAWA

## Ocena widoczności w powietrzu zamglonym metodą pomiaru transmisji światła monochromatycznego

### Evaluation of visibility in hazy air by measurement of the monochromatic light transmission

#### Streszczenie

W pracy rozpatrzono, jaki wpływ na ocenę widoczności, uzyskaną za pomocą pomiaru transmisji światła, ma zastosowanie źródła światła monochromatycznego. Dokonano analizy funkcji opisującej tłumienie światła i zmianę kontrastu w powietrzu zamglonym w zakresie widzialnym długości fal dla różnych wariantów obejmujących ocenę wzrokową, pomiar transmisji przy świetle białym wg definicji MOR i monochromatycznym. Wyprowadzono formuły, które umożliwiają obliczenie różnic w ocenie widoczności zależnie od sposobu pomiaru. Stwierdzono, że zastosowanie światła o długości fali 550 nm nie powoduje powstania istotnych różnic w stosunku do pomiaru MOR i oceny wzrokowej dokonanej w dzień. Dla innych długości fal zakresu widzialnego różnice te można zredukować przez zastosowanie opisanej w pracy procedury.

#### Abstract

Consequences of using monochromatic light source on visibility estimation made by light transmission is considered in the paper. An analysis of the function describing the light attenuation and contrast change within visible range in hazy and foggy air for several measurement procedures is performed. They include human visual estimation, measurement of white light transmission according to MOR definition and monochromatic light transmission.

The formulas are introduced, which enable to calculate differences in the visibility evaluation according to the procedure of measurement. It was found, that application the light of wavelength 550 nm is not the reason of essential differences in comparison with MOR measurement and daylight visual evaluation. For other wavelengths within range these differences can be reduced using the described here procedure.

#### 1. Wstęp.

Znaczenie pomiaru widoczności dla ogólnych celów synoptycznych i zastosowań meteorologii w komunikacji (zwłaszcza lotniczej) spowodowało rozwój metod umożliwiających ocenę tego parametru bez subiektywności, która towarzyszyła ocenom dokonywanym nawet przez dobrze wyszkolonego obserwatora. Niemniej odniesienie do istoty postrzegania wzrokowego oraz wykorzystanie światła jako czynnika mierzonego



jest podstawą działania licznych przyrządów do pomiaru zakresu widoczności, które są obecnie wytwarzane.

Jako miarę widoczności przyjmuje się horyzontalną odległość widzenia - pojęcie wprowadzone przez Koschmiedera (1924) i oznaczające maksymalny dystans, przy którym przedmiot staje się dla oka ludzkiego nieodróżnialny od tła. Parametr ten można ściślej określić jako odległość, przy której następuje spadek pierwotnego kontrastu między ciałem doskonale czarnym a tłem (horyzontem) do wartości progowej czułości wzroku. Zmiana kontrastu następuje w wyniku rozpraszania światła na cząstkach znajdujących się w atmosferze i jego absorpcji przez składniki atmosfery.

Bezpośrednie wykorzystanie powyższej definicji do pomiarów widoczności natrafia jednak na trudności związane ze zdalnym pomiarem luminancji obiektu i tła. Dlatego w praktyce stosuje się metody pośrednie polegające na pomiarze tłumienia (transmisji) strumienia światła na określonym odcinku jego drogi rozchodzenia się w badanym ośrodku lub pomiarze strumienia światła rozproszonego w danej objętości ośrodka [3]. Pomiary te umożliwiają określenie współczynnika ekstynkcji ośrodka, który charakteryzuje jego własności rozpraszające lub pochłaniające i jest stałą funkcji wykładniczej, zgodnie z którą wg prawa Bouguera - Lamberta światło ulega tłumieniu w ośrodku jednorodnym. Przyjmując, że kontrast zmienia się z tą samą stałą, można obliczyć odległość, przy której maleje on do wartości progowej.

Aby warunki pomiaru możliwie wiernie odpowiadały warunkom obserwacji wzrokowej, często narzuca się określone wymagania urządzeniu pomiarowemu i uściśla definicję zakresu widoczności. Zgodnie z zaleceniami Międzynarodowej Organizacji Meteorologicznej (1957) jest stosowany termin - meteorologiczny zasięg optyczny MOR (*Meteorological Optical Range*) - oznaczający odległość, przy której strumień światła wiązki skolimowanej wysyłanej przez lampę żarową o temperaturze barwowej 2700 K ulega zmniejszeniu do wartości 0,05, przy czym jest on mierzony za pomocą elementu o charakterystyce spektralnej odpowiadającej funkcji fotopowej podanej przez Międzynarodową Komisję Oświetleniową (CIE). Światło emitowane przez wskazane źródło odznacza się podobną charakterystyką spektralną co światło słoneczne na powierzchni Ziemi w zakresie widzialnym. Urządzenia oparte na podanej definicji zaliczane są do najbardziej wiarygodnych i dokładnych przyrządów do oceny zakresu widoczności.

Jednak oprócz urządzeń spełniających te wymagania stosuje się również inne, w których źródło światła promieniuje tylko wąskie pasmo długości fal z zakresu bliskiej podczerwieni lub ultrafioletu. Długookresowe badania porównawcze przeprowadzone w roku 1989 przez MOM potwierdziły ich przydatność dla celów ruchu lotniczego [5]. Pojawienie się laserów He - Ne umożliwiło powstanie przyrządów do pomiaru tłumienia światła w atmosferze przy użyciu światła monochromatycznego [1][4]. Szczególne możliwości dla budowy przyrządów służących do oceny widoczności stwarzają lasery półprzewodnikowe (diody laserowe), których rozwój doprowadził w ostatnich latach do seryjnej produkcji elementów wysyłających światło czerwone o długości fali 670 nm, a nawet 635 nm. Prowadzone są intensywne prace zmierzające do uzyskania jeszcze mniejszych długości fal, co pozwoliłoby na budowę w przyszłości urządzeń wykorzystujących światło o długości fali bliskiej tej, przy której czułość oka jest największa (550 nm). Taka długość fali przyjmowana jest również często do definicji zakresu widoczności w meteorologii i pracach badawczych nad atmosferą.



Lasery półprzewodnikowe odznaczają się ponadto wieloma innymi cechami korzystnymi dla budowy przyrządów do pomiaru widoczności, jak np. możliwość uzyskania skolimowanej wiązki o malej rozbieżności. Ułatwia to uzyskanie wysokich dokładności pomiaru, co pozwala z kolei na skrócenie odległości bazowej, na której dokonuje się pomiaru [11]. Transmisometry mogą być tańsze i łatwiejsze do zainstalowania niż tradycyjne urządzenia z lampą ksenonową.

Jednakże należałoby rozpatrzyć, jakie następstwa dla dokładności i wiarygodności oceny widoczności może mieć fakt, że pomiaru dokonuje się przy świetle monochromatycznym, za jakieś z punktu widzenia omawianych zastosowań można uważać światło emitowane przez diodę laserową. W szczególności celowe byłoby zbadanie, jakich różnic w wyniku pomiaru można oczekiwać między takim pomiarem, a pomiarem wykonanym zgodnie z zaleceniami MOM.

Problem ten zamierza się rozwiązać na drodze analizy funkcji tłumienia światła w powietrzu w obrębie pasma widzialnego dla przypadku postrzegania wzrokowego oraz pomiaru w świetle białym i monochromatycznym. Do tych obliczeń zostanie przyjęte, że charakterystyka spektralna atmosfery na powierzchni Ziemi może być opisana w postaci zależności znanych z literatury i opartych na wieloletnich badaniach. Ponieważ wyrażenia opisujące transmisję światła mają postać całkową, a występujące w nich niektóre składniki mogą być podane tylko w postaci zbiorów danych, zostanie dokonana analiza numeryczna tych wyrażań. Następnie zostanie zbadane, czy można określić zastępcze współczynniki ekstynkcji dla poszczególnych przypadków pomiaru transmisji światła. Posłuży to z kolei do sformułowania końcowych wniosków odnoszących się do możliwości wykorzystania pomiaru transmisji światła monochromatycznego do oceny widoczności.

## 2. Podstawa działania przyrządów doceny widoczności metodą pomiaru transmisji światła

W urządzeniu do oceny widoczności działającym na zasadzie pomiaru tłumienia światła w atmosferze (transmisometrze), niezależnie od możliwych wariantów jego budowy, można wyróżnić źródło światła (nadajnik) i umieszczony w pewnej odległości od niego detektor światła (odbiornik). Jeżeli drogę, którą przebywa światło oznaczmy przez  $x$ , to przezroczystość atmosfery może być przedstawiona za pomocą funkcji przepuszczania  $T(x)$  będącej stosunkiem strumienia światła mierzonego przez detektor w sytuacji, gdy występuje tłumienie światła w ośrodku, do strumienia mierzonego w warunkach braku takiego tłumienia.

Ze względu na to, że rozpraszanie i pochłanianie światła w atmosferze zależy od jego długości fali, a ponadto emisja i detekcja światła ma charakter spektralny, wyrażenie na funkcję przenoszenia przybiera w ogólnym przypadku postać całkową. Można je napisać biorąc pod uwagę prawo Bouguera-Lamberta, zgodnie z którym w ośrodku jednorodnym moc promieniowania o pewnej długości fali światła  $\lambda$  maleje wykładniczo w funkcji odległości od źródła ze stałą - współczynnikiem ekstynkcji  $\alpha(\lambda)$ . Jeżeli przez  $P(\lambda)$  oznaczmy moc wypromieniowaną przez źródło, a przez  $S(\lambda)$  czułość detektora, to dla mierzonego widma  $[\lambda_1, \lambda_2]$  otrzymamy następujące wyrażenie na funkcję  $T(x)$ :

$$(1) \quad T(x) = \frac{\int_{\lambda_1}^{\lambda_2} P(\lambda) S(\lambda) \exp[-\alpha(\lambda)] d\lambda}{\int_{\lambda_1}^{\lambda_2} P(\lambda) S(\lambda) d\lambda}$$

Wielkości  $P(\lambda)$  i  $S(\lambda)$  można przedstawić w postaci iloczynu wartości odniesienia i gęstości widmowej. Tak więc

$$(2) \quad T(x) = \frac{\int_{\lambda_1}^{\lambda_2} p(\lambda) s(\lambda) \exp[-\alpha(\lambda)] d\lambda}{\int_{\lambda_1}^{\lambda_2} p(\lambda) s(\lambda) d\lambda}$$

gdzie  $p(\lambda)$  i  $s(\lambda)$  oznaczają odpowiednio gęstości widmowe mocy promieniowania źródła i czułości detektora.

Podobną postać do powyższego wyrażenia ma wzór na zmianę kontrastu obiektu, jaka następuje w wyniku ograniczonej przeźroczystości atmosfery. Kontrast definiuje się jako stosunek luminancji ciała doskonale czarnego do luminancji tła, za które w interesującym nas przypadku przyjmuje się niebo przy horyzoncie. Zakładając, że ośrodek jest jednorodny i mierząc za pomocą fotometru luminancję obiektu oddalonego od niego o odległość  $x$  oraz luminancję tła otrzymuje się następujące wyrażenie na kontrast między nimi

$$(3) \quad C(x) = \frac{\int_{\lambda_1}^{\lambda_2} r(\lambda) n(\lambda) \exp[-\alpha(\lambda)x] d\lambda}{\int_{\lambda_1}^{\lambda_2} r(\lambda) n(\lambda) d\lambda}$$

W powyższym wzorze  $r(\lambda)$  oznacza gęstość widmową radiancji tła,  $n(\lambda)$  gęstość widmową czułości fotometru. Człon wykładniczy wynika z zastosowania prawa Bouguera-Lamberta do wyrażenia zmiany luminancji składowych widma o długości fali  $\lambda$ .

Pomiar kontrastu może posłużyć do określenia zakresu widoczności rozumianego zgodnie z pojęciem wprowadzonym przez Koschmiedera jako odległości, przy której kontrast spada do poziomu progowej czułości oka  $\varepsilon$  (przyjmowanej zwykle jako 0,02-0,05). Wiąże się to z prawem Webera-Fechnera, według którego widzenie polega na dostrzeganiu kontrastu luminancji, jeżeli obserwowany obiekt ma na tyle duże rozmiary, by jego obraz padał na wiele receptorów siatkówki oka. Horyzontalną odległość widzenia przy założeniu, że znany jest współczynnik ekstynkcji i funkcja  $n(\lambda)$  odpowiada charakterystyce czułości spektralnej oka, można by więc wyznaczyć ze wzoru (3) przez przyrównanie go do  $\varepsilon$ . Podobieństwo wzoru (3) do wyrażenia na funkcję przepuszczania (2) umożliwia wykorzystanie do tego samego celu funkcji  $T(x)$  pod warunkiem, że funkcja  $p(\lambda) s(\lambda)$  ma taki sam przebieg jak  $r(\lambda) n(\lambda)$ . Praktyczny pomiar spadku mocy promieniowania jest bowiem znacznie łatwiejszy do przeprowadzenia niż pomiar kontrastu. Takiej metodzie pomiaru widoczności odpowiada też przytoczona we wstępie definicja zasięgu optycznego MOR.

Jednakże analityczne wykorzystanie zależności (2) i (3) bez przyjęcia pewnych założeń upraszczających nie jest możliwe. Wynika to z charakteru funkcji występują-



cych w tych wzorach, które są zależnościami uzyskanymi na drodze pomiarowej i nie mają w zasadzie swojej reprezentacji analitycznej (wyjątkiem jest funkcja  $\alpha(\lambda)$ , którą przedstawia się za pomocą pewnego wzoru empirycznego).

W transmisometrach spełniających wymagania zawarte w definicji MOR wynik pomiaru jest obliczany przy założeniu, że wartość współczynnika ekstynkcji jest stała w zakresie widzialnym i równa pewnej średniej wartości  $\alpha_z$ . Wzory (2) i (3) upraszczają się wówczas do postaci:

$$T(x) = \exp(-\alpha_z x) \quad (4)$$

$$C(x) = \exp(-\alpha_z x). \quad (5)$$

Zasięg widoczności horyzontalnej  $V_z$  można obliczyć przyrównując  $T(x)$  do  $\epsilon$ , a zatem otrzymuje się

$$V_z = \frac{\ln \epsilon}{\alpha_z} \quad (6)$$

Wartość współczynnika  $\alpha_z$  oblicza się na podstawie pomiaru funkcji przenoszenia dla pewnej odległości bazowej  $B$ . Jeżeli więc  $T(B)$  oznaczmy przez  $T$ , to  $\alpha_z$  można obliczyć ze wzoru

$$\alpha_z = -\frac{\ln T}{B} \quad (7)$$

a po podstawieniu do wzoru (6) otrzymuje się

$$V_z = -\frac{B \ln \epsilon}{\ln T} \quad (8)$$

Zależności (2) i (3) upraszczają się do wzoru analogicznego do (6) również w przypadku, jeżeli pomiar odbywa się przy wykorzystaniu światła monochromatycznego o długości fali światła  $\lambda_k$ . Sytuacja taka ma miejsce w przypadku stosowania innych źródeł światła niż żarowe np.: laserów lub diod emitujących światło podczerwone o wąskim paśmie spektralnym. Otrzymujemy wówczas

$$T(x) = C(x) = \exp(-\alpha_k x), \quad (9)$$

gdzie  $\alpha_k$  oznacza współczynnik ekstynkcji dla długości fali świetlnej  $\lambda_k$ .

Współczynnik ekstynkcji i zasięg widoczności  $V_k$  zmierzony w tych warunkach można obliczyć jak poprzednio. Otrzymuje się wówczas

$$V_k = -\frac{\ln \epsilon}{\alpha_k}. \quad (10)$$

Przyjęcie omówionych wyżej założeń upraszczających może powodować powstanie różnicy oszacowania zakresu widoczności w stosunku do oceny wzrokowej. Jej przyczyną jest głównie zależność współczynnika ekstynkcji od długości fali. Dlatego to zagadnienie zostanie rozpatrzone przed przystąpieniem do dalszej analizy.

### 3. Wpływ długości fali światła na tłumienie światła w atmosferze

#### 3.1. Czynniki wpływające na tłumienie światła

Podczas propagacji światła w atmosferze jego energia ulega zmniejszeniu wskutek występowania następujących zjawisk: 1) absorpcji molekularnej, 2) rozpraszania molekularnego, i 3) rozpraszania aerozolowego [14]. Czynniki te oddziałują łącznie, a spowodowaną nimi wielkość tłumienia światła o długości fali  $\lambda$  można obliczyć ze wzoru

$$(11) \quad T_s = \exp[-\alpha_s(\lambda) x]$$

gdzie

$$(12) \quad \alpha_s = \alpha_{an}(\lambda) + \alpha_{m}(\lambda) + \alpha_{ra}(\lambda) ,$$

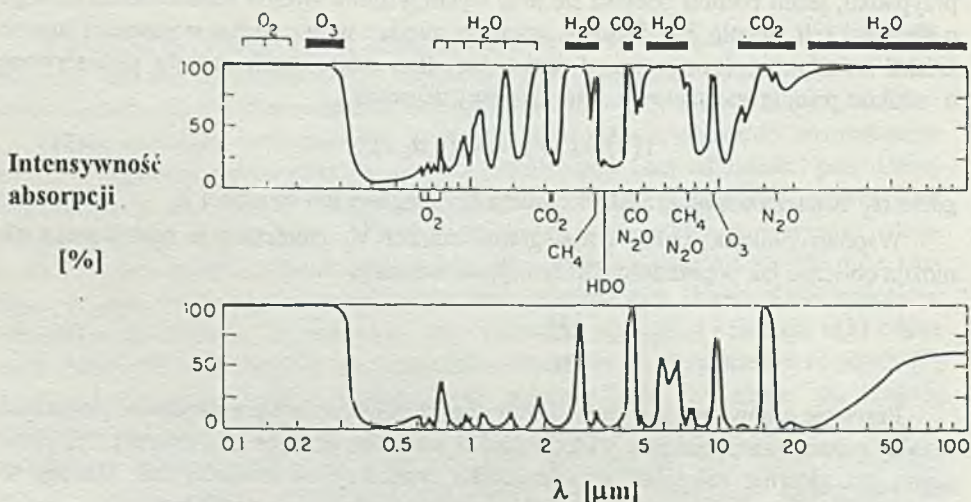
przy czym  $\alpha_s(\lambda)$  - sumaryczny współczynnik ekstynkcji,

$\alpha_{an}(\lambda)$  - współczynnik ekstynkcji dla absorpcji molekularnej,

$\alpha_m(\lambda)$  - współczynnik ekstynkcji dla rozpraszania molekularnego,

$\alpha_{ra}(\lambda)$  - współczynnik ekstynkcji dla rozpraszania aerozolowego.

Absorpcja molekularna jest spowodowana wpływem fali świetlnej na drgania cząstek znajdujących się w ośrodku. Do głównych składników atmosfery wywołujących absorpcję należą: para wodna, dwutlenek węgla, tlen, ozon, a także tlenek węgla, metan, tlenek azotu. Intensywność absorpcji zależy od długości fali światła i koncentracji danego gazu. Ogólny obraz pasm absorpcji pokazano na rys. 1.



Rys. 1. Intensywność absorpcji w funkcji długości fali światła [14] - spektrum promieniowania słonecznego na powierzchni Ziemi (górny wykres) i na wysokości 11 km (dolny wykres).



Absorpcja światła jak widać z podanych wykresów silnie zależy od długości fali światła. W pasmach charakteryzujących się dużym współczynnikiem absorpcji światło ulega całkowitemu stłumieniu już po przejściu kilku mm, a w przerwach między pasmami odległość ta osiąga setki metrów. Istnieją również centra słabej absorpcji, w których światło przechodzi na odległość do 1 km. Między nimi absorpcja światła nie ma istotnego znaczenia.

W interesującym nas zakresie widzialnym występują jedynie zakresy słabej absorpcji powodowanej przez ozon, tlen i parę wodną. Pierwszy z tych gazów poza nadfioletem i jego skrajem absorbuje światło w zakresie długości fal 450 - 740 nm. Oddziaływanie tlenu zaznacza się w pasmie, którego centrum przypada na 538,4 nm. Jeśli chodzi o parę wodną, to jej wpływ zaczyna się w końcowym obszarze pasma widzialnego i charakteryzuje się występowaniem licznych linii absorpcji oddalonych o ułamki nm.

**Rozpraszanie molekularne** na gazach znajdujących się w atmosferze ze względu na małe rozmiary molekul gazów ma charakter rozpraszania Rayleigha. Współczynnik ekstynkcji  $\alpha_{rm}$  zależy odwrotnie proporcjonalnie do czwartej potęgi długości fali. Wpływ tego zjawiska może więc wystąpić jedynie w dolnym przedziale pasma widzialnego, jednak i tutaj wartość tego współczynnika nie jest duża (ok. 0,04 /km, co odpowiada zakresowi widoczności 75 km).

**Rozpraszanie aerozolowe** następuje na cząstkach zawieszonych w atmosferze takich jak mgła, kurz, pyły przemysłowe, organizmy żywe itp. Ich średnica jest większa niż długość fali światła widzialnego lub co najwyżej z nią porównywalna, rozpraszanie światła na cząstkach aerozolu może być zatem opisane za pomocą wzorów wynikających z teorii Mie. Współczynnik ekstynkcji  $\alpha_{ra}$  zmienia się odwrotnie proporcjonalnie do potęgi długości fali o wykładniku  $q$ , którego wartość zależnie od średnicy cząstek aerozolu wynosi od ok. 0 do ok. 2 [13]. Dolna wartość jest osiągana, gdy średnica ta przekracza wielokrotnie długość fali światła, co występuje w przypadku np. deszczu, śniegu, gradu lub owadów; rozpraszanie praktycznie nie zależy wtedy od długości fali światła. Górna wartość występuje dla średnic cząstek znacznie mniejszych od długości fali światła, a charakter zjawiska staje się podobny do rozpraszania Rayleigha.

Jak wynika z przedstawionego opisu, wpływ poszczególnych zjawisk na tłumienie światła w atmosferze zależy od jej składu, co z kolei wiąże się z określonym stanem meteorologicznym. W skrajnym przypadku, gdy znajduje się w niej pomijalnie mało cząstek aerozolu (może to mieć miejsce np. w obszarach górskich lub po długich opadach), atmosferę określa się jako niemętną i wówczas odgrywa rolę rozpraszanie molekularne i absorpcja molekularna. Przy obecności mgieł i chmur dominującym zjawiskiem staje się z kolei rozpraszanie aerozolowe.

### 3.2. Zależność współczynnika ekstynkcji od długości fali światła przy dominującym wpływie rozpraszania aerozolowego

Złożony charakter zjawisk fizycznych będących przyczyną tłumienia światła w atmosferze, uzależnienie wielkości tłumienia od składu powietrza i aktualnych warunków meteorologicznych nie pozwala na sformułowanie ogólnych wyrażeń na zmianę współczynnika ekstynkcji w funkcji długości fali światła. Niemniej można stworzyć

reguly, które dla pewnych zastosowań praktycznych mogą opisywać tę zmianę z zadowalającą dokładnością.

Wydaje się, że takie rezultaty można osiągnąć w interesującym nas przypadku określania widoczności dla potrzeb komunikacji lotniczej lub drogowej, gdzie istotne jest monitorowanie stanów atmosfery charakteryzujących się niskimi i średnimi zakresami widoczności. Zgodnie z wymaganiami Międzynarodowej Organizacji Lotnictwa Cywilnego ICAO [6], parametr MOR powinien być wyznaczony z dokładnością lepszą niż  $\pm 10\%$  w zakresie 50 m + 1500 m., a powyżej tego przedziału z dokładnością  $\pm 20\%$ , przy czym wartości większe niż 10000 m nie są umieszczane w komunikatach meteorologicznych. Obniżenie widoczności do podanych wielkości jest powodowane głównie rozpraszaniem aerozolowym. Rozpraszanie molekularne nie wywołuje tak dużego spadku widoczności, również wpływ absorpcji molekularnej ze względu na występowanie w zakresie widzialnym jedynie słabych pasm absorpcji może być w pierwszym przybliżeniu pominięty.

Jak wspomniano wyżej, zmiana współczynnika ekstynkcji dla rozpraszania aerozolowego w funkcji długości fali światła może być wyrażona następująco

$$(13) \quad \alpha_{ra}(\lambda) \sim \lambda^{-q}$$

Wartość współczynnika  $q$  zależy od średnicy cząstek aerozolu, ich liczby i rozkładu wielkości. Do obliczeń przyjmuje się zwykle, że rozkład średnic cząstek jest rozkładem Junge'a tzn.

$$(14) \quad dn(r) = b r^{-(v+1)} dr,$$

gdzie  $dn(r)$  - liczba cząstek w jednostce objętości, których średnica mieści się w przedziale  $r$  i  $r+dr$ ,

$b$  - stała zależna od całkowitej liczby cząstek,

$v$  - parametr rozkładu.

Korzystając z tego wyrażenia można wykazać [7], że

$$(15) \quad \alpha_{ra}(\lambda) = C \lambda^{2-v} = C \lambda^{-q}$$

Parametry  $v$  i  $q = v - 2$  były określane przez licznych badaczy na drodze eksperymentalnej dla różnych stanów atmosfery i rodzajów aerozolu. Wyniki niektórych oszacowań przytoczono w pracy [13]. Junge ocenia, że dla typowego aerozolu kontynentalnego  $v$  wynosi 3, co oznaczałoby, że współczynnik ekstynkcji jest odwrotnie proporcjonalny do długości fali światła. Bullrich twierdzi, że wartość  $v$  dla naturalnego aerozolu może mieścić się w granicach 2,5 + 4. Middleton na podstawie długoletnich badań stwierdził, że parametr  $q$  przyjmuje wartość 1,3 przy zamgleniu, 1,6 przy zakresie widoczności 90 km i 2 przy widoczności 120 km, co oznaczałoby, że  $q = 2$  jest maksymalną wartością, której można oczekiwać. Zgodne z tym są wyniki badań Curcio, który stwierdził, że  $q$  zmniejsza się w przedziale od 0,7 do 1,9 przy zmianie zakresu widoczności od 2,8 km do 120 km. Z kolei Voltz na podstawie badań innych autorów podaje, że najczęściej występuje wartość  $q$  równa 1,2, a przedział zmian jest od 0,12 do 2,3. W podsumowaniu Woodman podaje, że jak wynika zarówno z teorii, jak i badań eksperymentalnych w zakresie widzialnym długości fal światła wartość współczynnika  $q$  wynosi średnio 1,3, a jej rozrzut  $\pm 0,6$ .



Z przytoczonych wyników widać, że wartość współczynnika  $q$  ulega zmianie zależnie od stanu atmosfery, który można z kolei scharakteryzować ogólnie za pomocą zakresu widoczności. Dla wartości tego parametru mniejszej lub równej 6 km, często przyjmuje się następującą zależność [8, 10]:

$$q = 0,585 V^{1/3} \quad (16)$$

W powyższym wzorze zakres widoczności  $V$ , wyrażony w km, jest zdefiniowany dla długości fali światła 550 nm, przy której czułość oka ludzkiego przy widzeniu dziennym jest najwyższa.

Występującą we wzorze (15) stałą  $C$  można również wyrazić za pomocą parametru  $V$ . Oznaczając przez  $\alpha(550)$  wartość współczynnika ekstynkcji dla długości fali światła 550 nm na podstawie wzoru (10) otrzymuje się

$$\alpha(550) = -\frac{\ln \varepsilon}{V} \quad (17)$$

a ponieważ wg wzoru (15)  $\alpha(550) = C(550)^{-q}$ , to obliczając stąd stałą  $C$  i podstawiając ją do wzoru (15) otrzymuje się

$$\alpha(\lambda) = -\frac{\ln \varepsilon}{V} \left( \frac{\lambda}{550} \right)^{-q} \quad (18)$$

Przyjmując wartość progu postrzegania wzrokowego tak, jak podano w definicji MOR (0,05) otrzymuje się

$$\alpha(\lambda) = \frac{3}{V} \left( \frac{\lambda}{550} \right)^{-q} \quad (19)$$

Powyższe wyrażenie przedstawia zmianę współczynnika ekstynkcji w funkcji długości fali światła w ośrodku, którego własności tłumiące światło są określone za pomocą zakresu widoczności horyzontalnej  $V$  mierzonej dla długości fali światła odpowiadającej maksymalnej czułości dziennego postrzegania wzrokowego. Występujący we wzorze parametr  $q$  w warunkach, gdy zakres ten nie przekracza kilku kilometrów, może być również wyrażony w funkcji  $V$  - zgodnie ze wzorem (16). Podstawiając go do wzoru (19) otrzymuje się

$$\alpha(\lambda) = \frac{3}{V} \left( \frac{\lambda}{550} \right)^{-0,585 V^{1/3}} \quad (20)$$

Tak więc dla potrzeb sygnalizowania stanów atmosfery, w których widoczność jest ograniczona, wzory (18) i (20) umożliwiają ocenę, w jakim stopniu tłumienie światła w atmosferze zależy od jego charakterystyki spektralnej.

## 4. Analiza funkcji przepuszczania światła i kontrastu

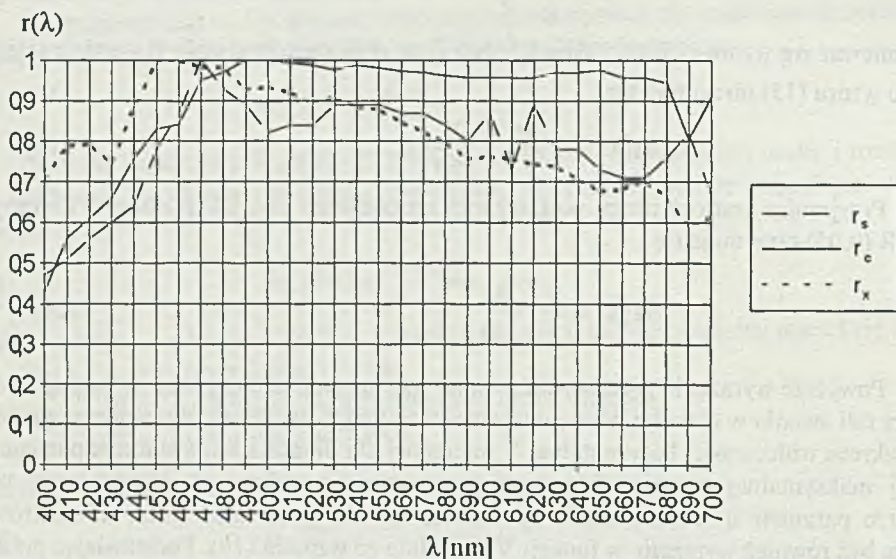
### 4.1. Metoda analizy

Celem dalszej analizy będzie zbadanie, w jakim stopniu będą różnić się wyniki oceny widoczności dokonane za pomocą pomiaru tłumienia światła białego i mono-

chromatycznego. Odniesieniem do takich porównań będzie ocena, która zostałaby dokonana przez hipotetycznego obserwatora ludzkiego charakteryzującego się pewnymi standardowymi cechami postrzegania wzrokowego.

Postępowanie takie przyjęto kierując się definicją widoczności podaną przez Koschmiedera. Na jej podstawie, jak podano to już uprzednio, horyzontalną odległość widzenia można otrzymać z równania (3) jako wartość zmiennej  $x$ , przy której przyjmuje ono wartość progowej czułości wzroku  $\varepsilon$ . Aby jednak dokonać takich obliczeń należy najpierw określić występujące w równaniu funkcje gęstości widmowej radiancji tła  $r(\lambda)$  i gęstości widmowej czułości wzroku  $\eta(\lambda)$ .

Jeśli chodzi o pierwszą z tych funkcji, to odpowiada ona charakterystyce spektralnej promieniowania słonecznego na powierzchni Ziemi. Dla celów niniejszej analizy przyjmijmy charakterystyki podane w literaturze [2], odnoszące się do dwóch sytuacji: braku chmur oraz pełnego zakrycia nieba chmurami. Przebieg tych charakterystyk dla zakresu widzialnego widma promieniowania zilustrowano na rys. 2.

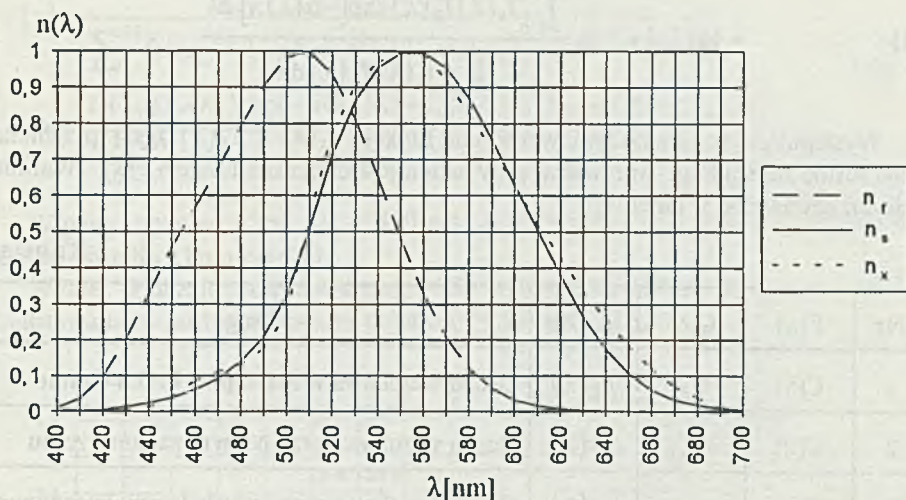


Rys. 2. Charakterystyka promieniowania słonecznego na powierzchni Ziemi przy braku chmur  $r_s$  i przy niebie całkowicie zakrytym chmurami  $r_c$  oraz charakterystyka spektralna promieniowania lampy ksenonowej  $r_x$

Na rysunku tym pokazano również gęstość widmową promieniowania lampy ksenonowej w obszarze widzialnym utworzoną na podstawie charakterystyki pokazanej w literaturze [9].

Co do funkcji  $n(\lambda)$ , to przyjmuje się, że jest nią krzywa widzialności fotopowa odnosząca się do widzenia dziennego, gdyż w zasadzie tylko wtedy można wzrokowo oceniać, jaki jest zasięg widoczności. Postrzeganie odbywa się wówczas za pomocą znajdujących się w siatkówce oka czopków. Charakterystykę widzialności dziennej zestandaryzowaną przez CIE pokazano na rys. 3. Maksimum czułości oka przypada dla długości fali światła 555 nm, spada ona natomiast praktycznie do zera przy 400 nm i 700 nm.





Rys. 3. Charakterystyki widmowe czułości oka: fotopowa  $n_f$  i skotopowa  $n_s$  oraz charakterystyka widmowa fotodetektora krzemowego  $n_x$

Jeżeli oświetlenie jest mniejsze niż  $10^{-2}$  lx (o zmierzchu), funkcję widzenia przejmują pręciki. Ich charakterystyka widmowa - skotopowa - jest przesunięta w lewo w stosunku do fotopowej i osiąga maksimum przy ok. 500 nm (rys 3). Granice widma widzialnego są podobne jak w poprzednim przypadku. Na rysunku pokazano również typową charakterystykę czułości fotodiody krzemowej z dodatkowym filtrem powodującym, że jest ona zbliżona do charakterystyki fotopowej [2].

Jeśli chodzi o występującą również we wzorze (3) funkcję spektralną współczynnika ekstynkcji  $\alpha(\lambda)$ , to została ona szczegółowo omówiona w poprzednim rozdziale

Jak wynika z przeprowadzonego omówienia funkcji występujących w wyrażeniu na zmianę kontrastu, zostały one utworzone na podstawie badań eksperymentalnych i nie mają (poza  $\alpha(\lambda)$ ) swojej postaci analitycznej. Tak więc określenie zasięgu widoczności horyzontalnej na podstawie wzoru (3) może odbyć się jedynie na drodze numerycznej.

Obliczenia takie przeprowadzono dla funkcji logarytmicznej wyrażenia na kontrast tzn. dla  $\ln C(x)$ , gdyż ułatwia to przeprowadzanie porównania między poszczególnymi wariantami pomiaru widoczności. Będą one obejmowały zarówno przypadek oceny wzrokowej, jak i opartej na pomiarze transmisji światła białego i monochromatycznego. W tym drugim przypadku zostanie wykorzystana funkcja przepuszczania (2), a przedmiotem obliczeń będzie  $\ln T(x)$ . Obliczenia zostaną przeprowadzone dla granic całkowania  $\lambda_1 = 400$  nm i  $\lambda_2 = 700$  nm, wynikających z omówionych wyżej przedziałów zmienności funkcji  $n(\lambda)$ .

#### 4.2. Obliczenia funkcji kontrastu i tłumienia światła.

Przedmiotem obliczeń będzie logarytm naturalny funkcji  $F(x)$  stanowiącej uogólnioną postać wyrażen na  $C(x)$  i  $T(x)$

$$(21) \quad -\ln F(x) = -\ln \frac{\int_{400}^{700} f_1(\lambda) f_2(\lambda) \exp[-\alpha(\lambda)x] d\lambda}{\int_{400}^{700} f_1(\lambda) f_2(\lambda) d\lambda}$$

Występujące w powyższym wyrażeniu funkcje  $f_1(\lambda)$  i  $f_2(\lambda)$  będą przybierały różną formę zależnie od warunków, przy jakich jest obliczana funkcja  $F(x)$ . Warianty obliczeń zestawiono w tabeli 1.

Tabela 1

Nr	$F(x)$	$f_1(\lambda)$	$f_2(\lambda)$	Uwagi
1	$C(x)$	$r_s(\lambda)$	$n_f(\lambda)$	Ocena wzrokowa w dzień przy braku chmur
2	$C(x)$	$r_c(\lambda)$	$n_f(\lambda)$	Ocena wzrokowa przy pełnym zachmurzeniu
3	$C(x)$	$r_x(\lambda)$	$n_f(\lambda)$	Ocena wzrokowa przy świetle lampy ksenonowej
4	$C(x)$	$r_s(\lambda)$	$n_s(\lambda)$	Ocena wzrokowa o zmierzchu przy braku chmur
5	$C(x)$	$r_c(\lambda)$	$n_s(\lambda)$	Ocena wzrokowa o zmierzchu przy pełnym zachmurzeniu
6	$T(x)$	$r_c(\lambda)$	$n_d(\lambda)$	Pomiar transmisji, światło słoneczne, pełne zachmurzenie
7	$T(x)$	$r_x(\lambda)$	$n_d(\lambda)$	Pomiar transmisji, światło lampy ksenonowej

W obliczeniach przyjęto, że  $\alpha(\lambda)$  zmienia się zgodnie równaniem (20). Występujący w nim parametr  $V$  obrano z zakresu 100 m do 10000 m obejmującego sytuacje najbardziej istotne dla ruchu samochodowego i lotniczego tj. od bardzo ograniczonej widoczności spowodowanej silną mgłą do dobrej przy słabym zamgleniu. Główną przyczyną ograniczenia widoczności jest wtedy rozpraszanie aerozolowe i dopuszczalne jest przyjęcie założeń upraszczających, przy których określono zależność współczynnika ekstynkcji od długości fali światła.

Obliczenia wykonano dla wartości współczynnika  $\epsilon$  równej 0,05, gdyż zgodnie z definicją meteorologicznego zasięgu optycznego MOR jest ona przyjmowana powszechnie w stosowanych obecnie transmisometrach.

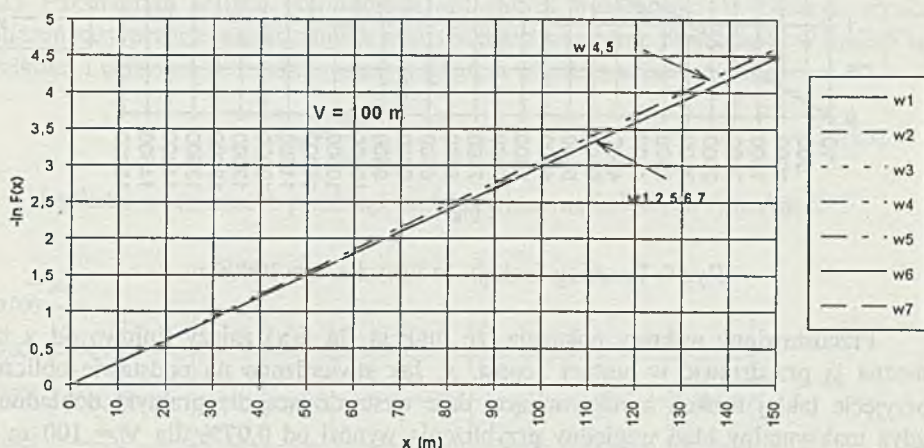
Funkcję (21) obliczono numerycznie po zastąpieniu występujących w niej całek szeregiem, które wyznaczono metodą trapezów. Aproksymacji dokonano przy podzieleniu zakresu całkowania na 31 odcinków o długości 10 nm. Ostatecznie więc przedmiotem obliczeń stało się wyrażenie  $-\ln F(x)$ , gdzie  $F(x)$  jest określone równaniem :



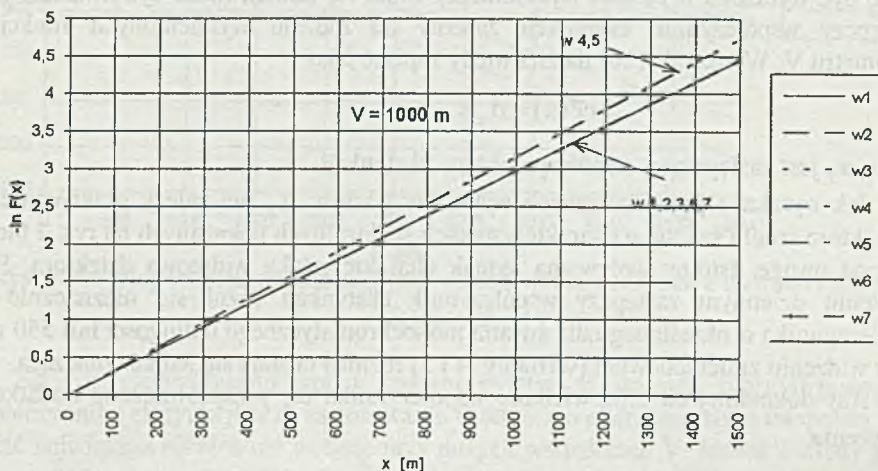
$$F(x) = \frac{\sum_1^{31} f_1(\lambda_k) f_2(\lambda_k) \exp[-\alpha(\lambda_k)x]}{\sum_1^{31} f_1(\lambda_k) f_2(\lambda_k) - 0,5[f_1(\lambda_1) f_2(\lambda_1) + f_1(\lambda_{31}) f_2(\lambda_{31})]} - 0,5 \frac{f_1(\lambda_1) f_2(\lambda_1) \exp[-\alpha(\lambda_1)x] + f_1(\lambda_{31}) f_2(\lambda_{31}) \exp[-\alpha(\lambda_{31})x]}{\sum_1^{31} f_1(\lambda_k) f_2(\lambda_k) - 0,5[f_1(\lambda_1) f_2(\lambda_1) + f_1(\lambda_{31}) f_2(\lambda_{31})]} \quad (22)$$

W powyższym wyrażeniu  $\lambda_1 = 400$  nm,  $\lambda_{31} = 700$  nm, a  $\lambda_k$  oznacza wartości długości fali światła z tego zakresu.

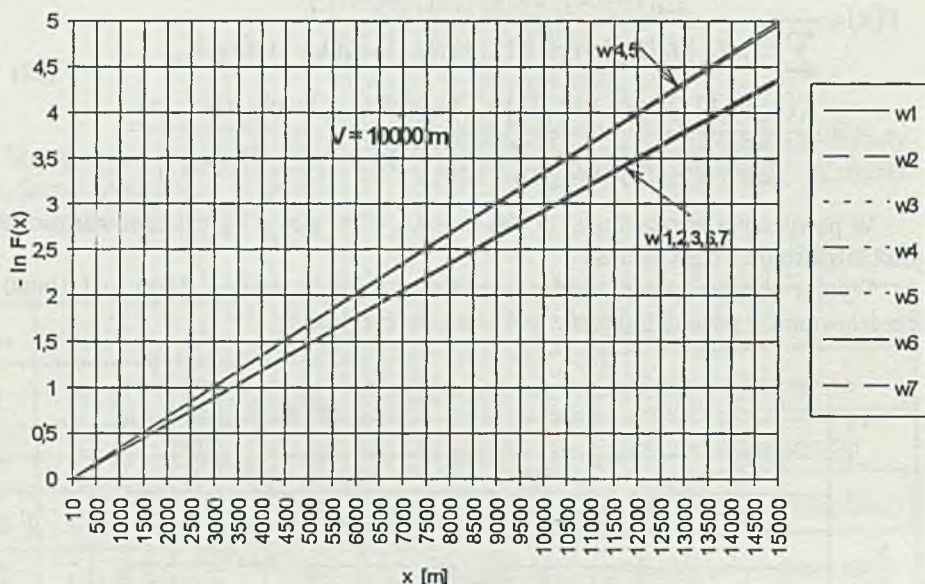
Wyniki obliczeń przeprowadzonych dla  $V$  równych 100 m, 1000 m i 10000 m przedstawiono w postaci wykresów pokazanych na rys. 4+6.



Rys. 4. Wykres zmian funkcji  $-\ln F(x)$  dla  $V = 100$  m.



Rys.5. Przebieg funkcji  $-\ln F(x)$  dla  $V = 1000$  m



Rys 6. Przebieg funkcji  $-\ln F(x)$  dla  $V = 10000$  m.

Przedstawione wykresy pokazują, że funkcja  $-\ln F(x)$  zależy liniowo od  $x$  tzn. można ją przedstawić w postaci  $\text{const.} \cdot x$ . Jak stwierdzono na podstawie obliczeń, przyjęcie takiej funkcji aproksymującej daje wystarczającą dla praktyki dokładność, gdyż maksymalny błąd względny przybliżenia wynosi od 0,07% dla  $V = 100$  m do 1,3% dla  $V = 10000$  m.

Wynika stąd, że skomplikowane wyrażenie na zmianę kontrastu  $C(x)$  lub tłumienia światła  $T(x)$  dla interesujących nas przypadków zmienności funkcji  $f_1(\lambda)$  i  $f_2(\lambda)$  może być wyrażone w postaci wykładniczej. Stała tej funkcji może być uważana jako zastępczy współczynnik ekstynkcji zależny od rodzaju wymienionych funkcji i parametru  $V$ . Wyrażenie (20) można wtedy zapisać jako

$$(23) \quad -\ln F(x) = \alpha_z \cdot x$$

gdzie  $\alpha_z$  jest zastępczym współczynnikiem ekstynkcji.

Jak wynika z przedstawionych wykresów wartość  $\alpha_z$  nie zależy praktycznie od tego, które źródło światła o charakterystykach spektralnych pokazanych na rys. 2 bierze się pod uwagę. Istotny wpływ ma jednak charakterystyka widmowa detektora. Przy widzeniu dziennym zastępczy współczynnik ekstynkcji różni się nieznacznie od współczynnika  $\alpha$  określonego dla światła monochromatycznego o długości fali 550 nm. Przy widzeniu zmierzchowym (warianty 4 i 5) różnica ta staje się jednak znacząca.

Aby dokładniej określić wielkość współczynnika  $\alpha_z$  przeprowadzono dodatkowe obliczenia.



### 4.3 Obliczenie zastępczego współczynnika ekstynkcji

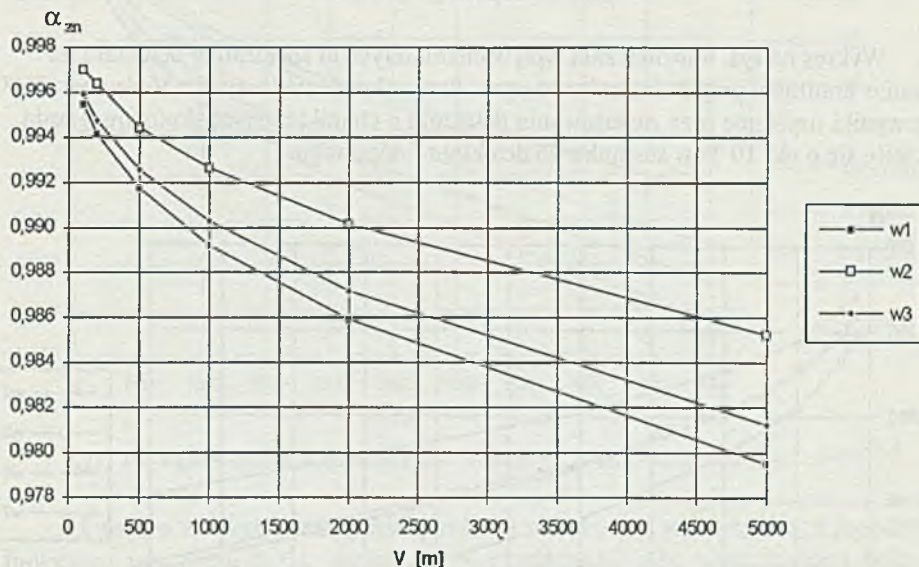
Duża liniowość funkcji  $\ln F(x)$  umożliwia aproksymację tej funkcji za pomocą prostej przechodzącej przez początek układu współrzędnych i punkt, w którym wartość tej funkcji wynosi 3. Odcięta tego punktu oznacza zakres widoczności horyzontalnej  $V_z$ , która jest mierzona w warunkach określonych metodą pomiaru (kontrastu lub transmisji światła) oraz rodzajem źródła światła i detektora (warianty 1 - 7). Nachylenie prostej stanowi zastępczy współczynnik ekstynkcji, który wynosi zatem

$$\alpha_z = \frac{3}{V_z} \quad (24)$$

Wielkość  $V_z$  może być obliczona numerycznie przy wykorzystaniu wyrażenia (22). Parametrem w tych obliczeniach jest zakres widoczności  $V$ . Dlatego wyniki obliczeń zastępczego współczynnika ekstynkcji jest wygodnie przedstawić w funkcji tej wielkości i w formie znormalizowanej względem współczynnika  $\alpha$  tj. dla

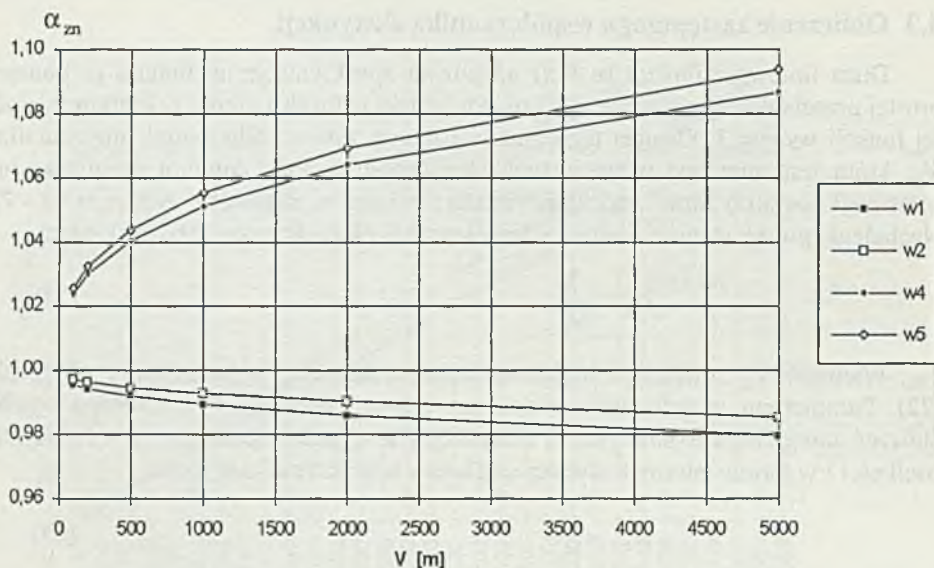
$$\alpha_{zn} = \frac{\alpha_z}{\alpha} \quad (25)$$

Uzyskane w ten sposób przebiegi przedstawiono na kolejnych rysunkach.



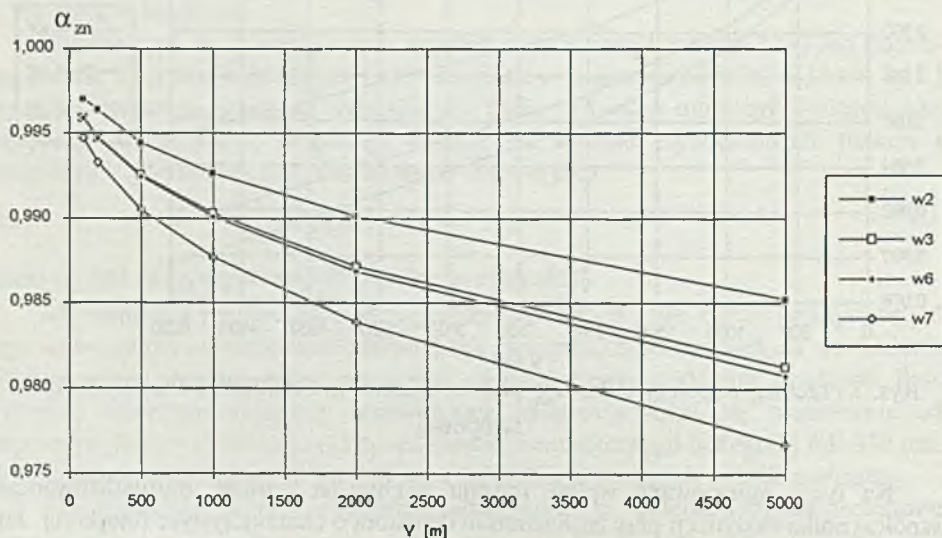
Rys. 7 Przebieg współczynnika  $\alpha_{zn}$  przy zastosowaniu detektora o charakterystyce fotonowej

Na rys.7 zobrazowano wpływ rodzaju światła na wartość znormalizowanego współczynnika ekstynkcji przy zastosowaniu detektora o charakterystyce fotonowej. Jak widać największa różnica ma miejsce przy dużych wartościach  $V$ . Jednak i wtedy wzajemne różnice są rzędu 0,5%, a w stosunku do  $\alpha$  zastępczy współczynnik ekstynkcji różni się o ok. 2%.



Rys. 8 Przebieg współczynnika  $\alpha_{zn}$  przy zastosowaniu detektora fotopowego (w1 i w2) oraz skotopowego (w4 i w5)

Wykres na rys. 8 uwidacznia wpływ charakterystyki spektralnej detektora na pomiar kontrastu przy świetle słonecznym. Przy zakresie widoczności  $V$  równym 5000 m, wyniki uzyskane przy zastosowaniu detektora o charakterystyce skotopowej będą różniły się o ok. 10 % w stosunku do detektora fotopowego.



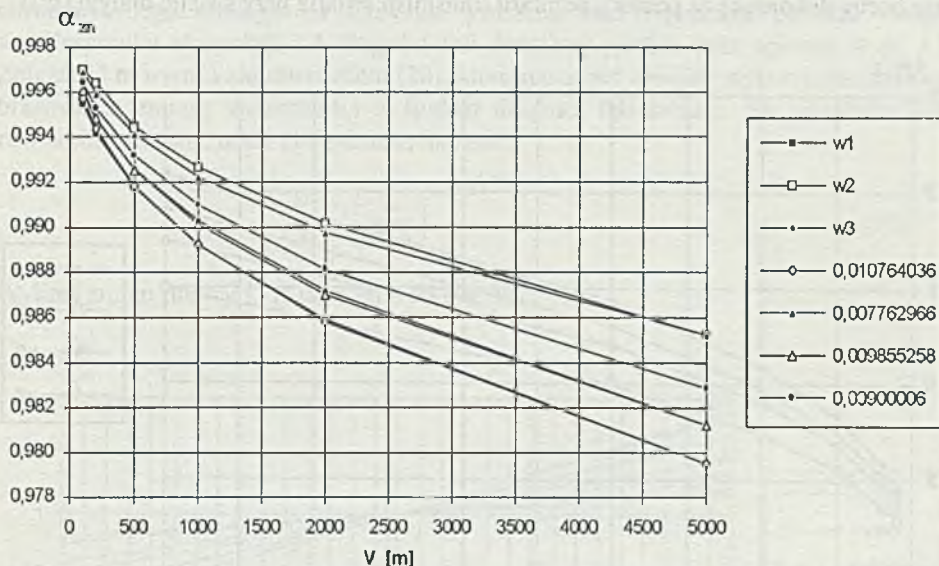
Rys. 9 Przebieg współczynnika  $\alpha_{zn}$  przy pomiarze kontrastu (w2 i w3) oraz transmisji światła (w6 i w7)



Na rysunku 9 zestawiono przebiegi współczynnika  $\alpha_{zn}$  dla porównania oceny dokonywanej za pomocą detektora fotopowego przy świetle słonecznym (w2 - niebo zachmurzone) i świetle lampy ksenonowej (w3) oraz detektora o charakterystyce zbliżonej do fotopowej i przy takich samych rodzajach światła (w6 i w7). Jak widać zmiana detektora powoduje różnice ok. 0,5 %, podobny jest też wpływ źródła światła. Wszystkie pokazane przypadki odznaczają się podobną zależnością współczynnika  $\alpha_{zn}$  od zakresu widoczności V. Nasuwa to przypuszczenie, że jest możliwe napisanie wzoru, który by podawał tę zależność analitycznie. Próby zasymulowania tych przebiegów funkcjami wykładniczymi pokazały, że dostatecznie dokładne wyniki dla przypadku detektora fotopowego uzyskuje się przy zastosowaniu wzoru

$$\alpha_{zn} = 1 - kV^{0,4} \quad (26)$$

gdzie k jest stałą zależną od rodzaju źródła światła i detektora. Dla wartości k równej 0,009 uzyskuje się przebieg średni, który pokazano na rys. 10 wraz z innymi krzywymi aproksymującymi przebiegi w1 - w3.



Rys. 10 Aproksymacja przebiegów funkcji  $\alpha_{zn}$  za pomocą funkcji  $1 - kV^{0,4}$

Jak można zauważyć maksymalna różnica między tym przebiegiem, a pozostałymi funkcjami jest rzędu 0,3%; można ją więc traktować jako wystarczająco dokładne przybliżenie tych funkcji.

## 5. Ocena widoczności przy zastosowaniu światła monochromatycznego

Opisane w poprzednim rozdziale obliczenia i wykresy pokazują wpływ, jaki mają na wynik pomiaru transmisji światła charakterystyki spektralne źródła światła i detektora. Wpływ ten może być przedstawiony za pomocą zastępczego współczynnika ekstynkcji, który jest również funkcją stanu atmosfery. Przy wykorzystaniu tego współczynnika można obliczyć, jakie różnice wystąpią w wynikach pomiaru zakresu

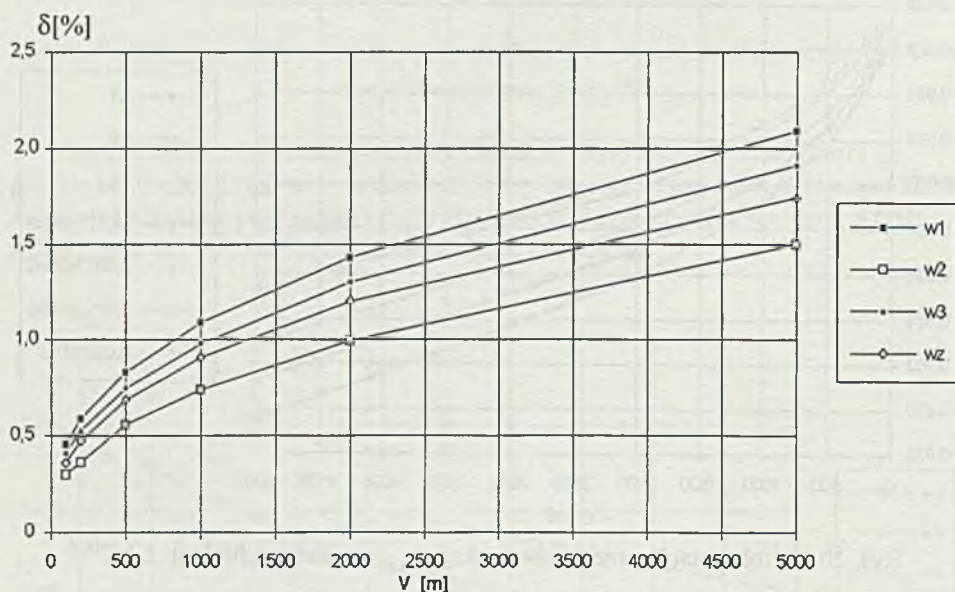
widoczności dla różnych wariantów pomiarowych. Różnice te jest wygodnie odnieść do zakresu widoczności określonego za pomocą pomiaru transmisji światła monochromatycznego o długości fali 550 nm tj. wielkości  $V$ . Wówczas względna różnica zakresu widoczności  $\delta$  wyniesie

$$(27) \quad \delta = \frac{(V_z - V)}{V}$$

gdzie  $V_z$  oznacza zakres widoczności odpowiadający współczynnikowi ekstynkcji  $\alpha_z$ . Korzystając ze wzorów (6), (8) i (26) otrzymuje się

$$(28) \quad \delta = \frac{1 - \alpha_m}{\alpha_m}$$

Podstawiając do tego wzoru obliczone uprzednio dla różnych wariantów wartości  $\alpha_m$  można przedstawić  $\delta$  w funkcji  $V$ . Na rysunku 11 przedstawiono takie przebiegi dla przypadku oceny wzrokowej widoczności dokonywanej w dzień (warianty w1 i w2) oraz oceny dokonanej za pomocą pomiaru transmisji światła przy świetle białym (w3) oraz przebieg zastępczy dla tych przypadków (wz).



Rys.11. Wykres zmian  $\delta$  przy ocenie wzrokowej widoczności (w1 i w2) i jej pomiarze przy świetle białym (w3) oraz przebieg zastępczy dla tych przypadków (wz).

Jak wynika z rysunku, przy widoczności 5000 m różnica między wynikiem pomiaru widoczności uzyskanym przy użyciu światła monochromatycznego 550 nm i światła białego wynosi od 1,5 do ok. 2 %, a przy widoczności obniżonej jest ona poniżej 1 %. Z rysunku wynika również, że różnice między wynikami uzyskanymi dla wymienionych wariantów wynoszą ok. 0,5 %, co nie ma praktycznego znaczenia. Przy wykorzystaniu zależności (26) można te przebiegi przedstawić w przybliżeniu za pomocą zależności



$$\delta = \frac{kV^{0,4}}{1 - kV^{0,4}} \quad (29)$$

Wykres zmian tej funkcji dla  $k = 0,009$ , przedstawiony również na rys. 11 jako wz, pokazuje, że zadowalająco odtwarza on charakter omawianych przebiegów. Jeszcze z większą dokładnością można aproksymować poszczególne przebiegi przyjmując wartości  $k$  podane na rys. 10.

Wprowadzone pojęcia i omówione zależności pozwalają na łatwe przeliczanie wyników uzyskanych za pomocą transmisometru, w którym wykorzystuje się światło monochromatyczne o długości fali 550 nm na wielkość zgodną np. z definicją MOR (odpowiada jej wariant w3). Przekształcając wyrażenie (25) otrzymuje się wzór

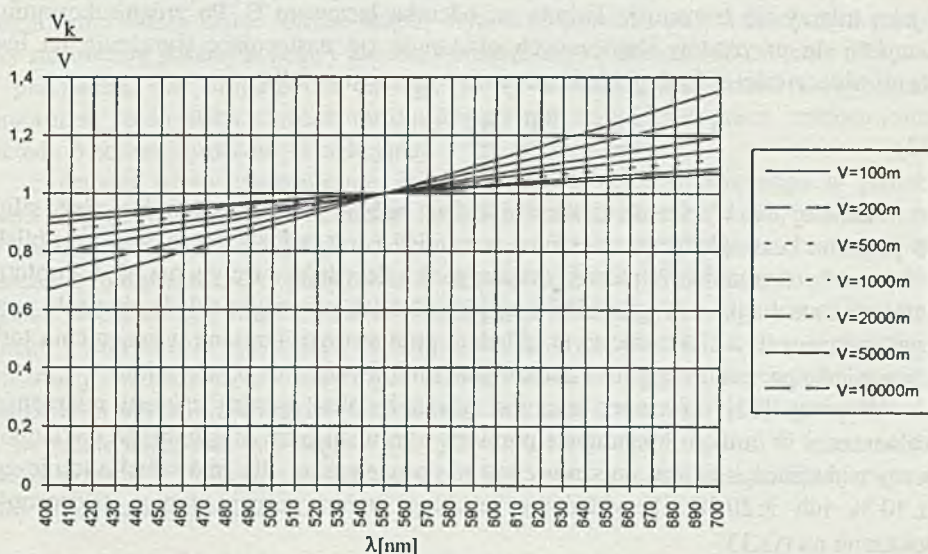
$$V_z = V(1 + \delta) \quad (30)$$

który umożliwia dokonanie takich przeliczeń po podstawieniu wartości  $\delta$  wziętej z pokazanych wykresów lub obliczonej ze wzoru (29).

Dodatkowa poprawka musi być jednak przyjęta, jeżeli długość fali światła monochromatycznego odbiega od 550 nm. Potrzeba taka wynika z powodu zmiany współczynnika ekstynkcji od długości fali świetlnej, która była opisana w p. 3.2. Zależność ta wyraża się równaniem (20), które może być również wykorzystane do zobrazowania zmiany widoczności w funkcji długości fali światła. Na podstawie tego równania oraz zależności (10) można napisać

$$\frac{V_k}{V} = \left( \frac{\lambda}{550} \right)^{0.585V^{1/3}} \quad (31)$$

Wykres zmian funkcji (31) pokazano na rys. 12.



Rys. 12. Wykres zmian zakresu widoczności w funkcji długości fali świetlnej

Tak więc zależnie od tego, czy długość fali światła, przy której dokonuje się pomiaru transmisji światła, jest mniejsza czy też większa niż 550 nm, wartość wyniku pomiaru widoczności będzie mniejsza lub większa od wartości  $V$ . Różnica zależy przy tym od stopnia tłumienia światła w atmosferze - maleje, jeżeli jest ono duże.

Równanie (31) pozwala na obliczenie wielkości  $V$ , jeżeli jest znany wynik pomiaru  $V_k$  dokonany przy określonej długości fali świetlnej  $\lambda_k$ . Otrzymuje się z niego wyrażenie

$$(32) \quad V = V_k \left( \frac{\lambda_k}{550} \right)^{-0,585 V^{1/3}}$$

Ponieważ  $V$  występuje w tym wzorze w postaci uwiklanej, praktyczne obliczenia można przeprowadzić po dokonaniu pewnych uproszczeń polegających na wprowadzeniu do pierwiastka znajdującego się w wykładniku potęgi zamiast  $V$  wielkości  $V_k / w$ , przy czym współczynnik  $w$  można określić z wykresu podanego na rys. 11 dla danej długości fali jako wartość przebiegu średniego (np. dla  $V = 1000$  m). Powstający z tego tytułu błąd dla pasma widzialnego nie ma praktycznego znaczenia. Np. dla źródła światła lasera diodowego o długości fali  $\lambda = 670$  nm, wartość  $w$  wynosi ok. 1,15, a błąd aproksymacji zmienia się od 0,15 % dla  $V = 100$  m do 0,9 % dla  $V = 10000$  m.

Przy omawianiu sposobu przeliczania wyników pomiaru uzyskanych przy stosowaniu światła monochromatycznego na wartości odpowiadające pomiarowi przy świetle białym i związanych z tym błędów oszacowania należy również zwrócić uwagę na inne błędy pomiaru, które mają miejsce przy ocenie widoczności metodą pomiaru tłumienia światła.

Jak wynika ze wzoru (8), dokładność takiej oceny jest uzależniona od dokładności z jaką mierzy się transmisję światła na odcinku bazowym  $B$ . Po zróżniczkowaniu i przejściu do przyrostów skończonych otrzymuje się następujące wyrażenie na błąd oceny widoczności w funkcji błędu pomiaru transmisji światła

$$(33) \quad \frac{\Delta V}{V} = - \frac{1}{\ln T} \frac{\Delta T}{T}$$

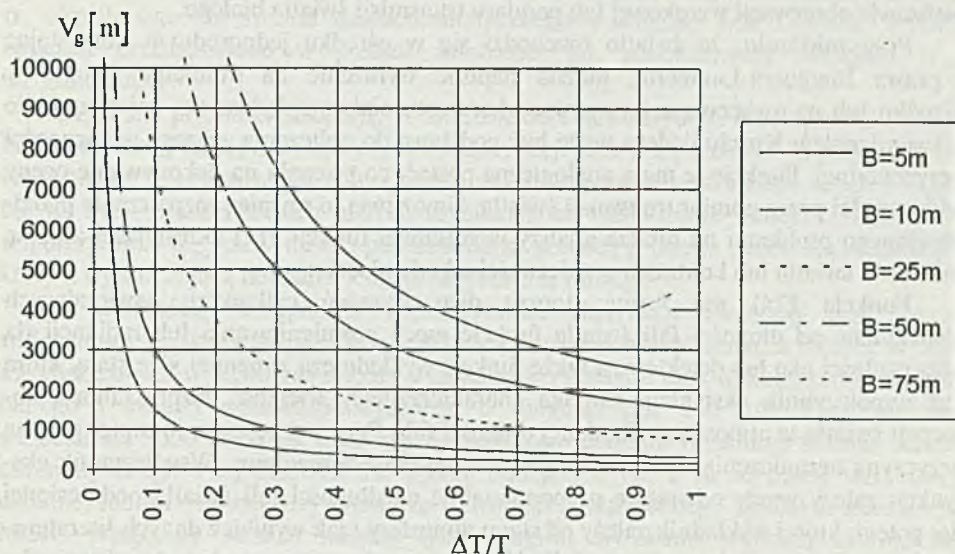
Analizując to wyrażenie można stwierdzić, że błąd oceny widoczności zależy silnie od poziomu bezwzględnego mierzonej transmisji i jest największy, gdy jest on bliska 1 lub też bardzo mała. W pierwszym przypadku decydujący wpływ ma błąd absolutny pomiaru transmisji, w drugim błąd względny. Na ich powstanie składa się wiele przyczyn związanych z elektronicznym układem pomiarowym (zmiana wzmocnienia toru, przesunięcie poziomu sygnału) i optycznym (zanieczyszczenie powierzchni).

W pracy [12] pokazano, jaka jest górna i dolna granica zakresu mierzonych widoczności w funkcji dokładności pomiaru transmisji przy założeniu, że dokładność oceny widoczności nie jest gorsza niż jest to wymagane np. dla celów ruchu lotniczego ( $\pm 10$  % lub  $\pm 20$  %). Przykładowy przebieg zmian górnej granicy widoczności pokazano na rys. 13.

Jak wynika z powyższego rysunku uzyskanie dokładności oceny widoczności równej 10 % wymaga stosowania aparatury zapewniającej dokładność pomiaru transmisji przynajmniej 1 %, jeżeli zakres pomiaru widoczności ma sięgać 2000 m. Jest to możliwe przy tym w przypadku, gdy odległość bazowa wynosi 75 m. Przy mniejszych



bazach rosną wymagania co do dokładności pomiaru transmisji. Z kolei duża baza uniemożliwia monitorowanie z odpowiednią dokładnością bardzo dużego zamglenia.



Rys.13 .Zależność górnej granicy zakresu widoczności od dokładności pomiaru transmisji światła przy warunku  $\Delta V / V = 10 \%$ .

Standardowe transmisometry charakteryzują się na ogół dokładnością pomiaru transmisji rzędu  $\pm 1\%$  [5]. Ocena widoczności jest więc zarówno ze względów technicznych, jak i natury tego parametru obciążona dużymi błędami. W tym aspekcie za drugorzędne należy uznać różnice między wynikami pomiaru widoczności uzyskanymi przy stosowaniu światła białego i monochromatycznego o długości fali światła 550 nm. W przypadku, gdy długość fali odbiega od tej wartości, rozbieżności w wynikach pomiaru są rzędu kilku procent; można je przy tym zmniejszyć przez zastosowanie uprzednio podanej procedury korekcyjnej.

Z drugiej strony zastosowanie źródła światła monochromatycznego w postaci lasera półprzewodnikowego daje szereg korzyści. Światło może być łatwo modulowane za pomocą napięcia sterującego, a moc promieniowania stabilizowana w szerokich zakresach temperatur. W połączeniu z odpowiednim układem optycznym można uzyskać skolinowaną wiązkę światła o małej rozbieżności. Rozmiary lasera i sprzężonego z nim układu optycznego są bardzo małe, co pozwala na uzyskanie zwartych konstrukcji urządzenia pomiarowego. Dobrze określone parametry wiązki światła umożliwiają uzyskanie wysokich dokładności pomiaru transmisji światła, co pozwala na skrócenie odległości bazowej, na której dokonuje się tego pomiaru. W wyniku tego można rozszerzyć dolną granicę zakresu pomiarowego oraz zmniejszyć obszar zajmowany przez transmisometr.

## 6. Podsumowanie i wnioski

Głównym celem pracy było zbadanie, jakie konsekwencje dla dokładności i wiarygodności oceny widoczności dokonanej metodą pomiaru transmisji światła ma



zastosowanie źródła światła monochromatycznego, a w szczególności określenie jaka istnieje korelacja między wynikami uzyskanymi tą drogą a ocenami otrzymanymi na podstawie obserwacji wzrokowej lub pomiaru transmisji światła białego.

Przy założeniu, że światło rozchodzi się w ośrodku jednorodnym korzystając z prawa Bouguera-Lamberta, można napisać wyrażenie na tłumienie światła w ośrodku lub na związaną z tym zmianę kontrastu ciała względem tła, która zgodnie z twierdzeniem Koschmiedera może być podstawą do obliczenia zakresu widoczności horyzontalnej. Funkcje te mają analogiczną postać, co pozwala na dokonywanie oceny widoczności przez pomiar transmisji światła. Umożliwia to również rozpatrzenie przedstawionego problemu na drodze analizy uogólnionej funkcji  $F(x)$  określającej zmianę tłumienia światła lub kontrastu w zależności od odległości  $x$ .

Funkcja  $F(x)$  ma formę ilorazu dwu wyrażeń całkowitych zawierających uzależnione od długości fali światła funkcje mocy promieniowania lub radiancji tła oraz czułości oka lub detektora, a także funkcję wykładniczą zmiennej  $x$  ze stałą, którą jest współczynnik ekstynkcji ośrodka charakteryzujący wielkość rozpraszania i absorpcji światła w atmosferze dla danej długości fali. Przy występowaniu mgieł główną przyczyną ograniczenia widoczności jest rozpraszanie aerozolowe. Współczynnik ekstynkcji zależy wtedy odwrotnie proporcjonalnie od długości fali światła podniesionej do potęgi, której wykładnik zależy od stanu atmosfery i jak wynika z danych literaturowych, można przyjąć, że jego wielkość jest wprost proporcjonalna do pierwiastka trzeciego stopnia z zakresu widoczności  $V$ , zmierzonego dla długości fali światła 550 nm.

Obliczenie funkcji  $F(x)$  ze względu na to, że występujące w niej funkcje dotyczące źródła i detektora światła mają formę wykresów uzyskanych na drodze pomiarowej, może mieć miejsce jedynie na drodze numerycznej. Obliczenia takie, po zastąpieniu całek szeregiem określonymi metodą trapezów, przeprowadzono dla 7 wariantów obejmujących przypadki obserwacji wzrokowej w dzień i o zmierzchu, przy niebie wolnym od chmur i w pełni pokrytym chmurami oraz pomiaru transmisji światła przy spełnieniu warunków podanych w definicji zasięgu widoczności MOR. Do obliczeń wykorzystano podane w literaturze charakterystyki promieniowania słonecznego na powierzchni Ziemi, charakterystyki fotopowe i skotopowe czułości oka, a także typowe charakterystyki promieniowania lampy ksenonowej i detektora krzemowego, stosowanych w transmisometrach do pomiaru MOR. Obliczenia przeprowadzono dla funkcji  $-\ln F(x)$  dla pasma widzialnego 400 nm - 700 nm, podzielonego na 31 odcinków. Jako parametr przyjęto wielkość  $V$  z zakresu 100 m do 10000 m.

Uzyskane tą drogą przebiegi funkcji  $-\ln F(x)$  okazały się liniowymi dla wszystkich rozpatrywanych wariantów, co umożliwia jej aproksymację za pomocą prostej przechodzącej przez początek układu współrzędnych i punkt, w którym wartość równa jest 3 tj. logarytmowi naturalnemu wartości progowej czułości oka, przyjętej w definicji MOR. Błąd względny takiego przybliżenia jest pomijalny, gdyż w rozpatrywanych zakresach widoczności wynosi on 0,1÷1 %. Tangens kąta nachylenia prostej można traktować jako zastępczy współczynnik ekstynkcji  $\alpha_z$  odnoszący się do określonych warunków pomiarowych i uwzględniający udział wszystkich składowych widma. Jak wynika z uzyskanych przebiegów największy wpływ na wartość współczynnika  $\alpha_z$  mają charakterystyki spektralne czułości oka, a znacznie mniej zależy on od rodzaju źródła światła. Wynika to stąd, że w interesujących nas przypadkach moc promieniowania w przeciwieństwie do tych charakterystyk zmienia się stosunkowo niewiele w paśmie wi-

działnym. Aby jednak dokładniej przedstawić ten wpływ, przeprowadzono dokładne obliczenia współczynnika  $\alpha_z$  znormalizowanego względem współczynnika ekstynkcji  $\alpha$ , określonego dla światła monochromatycznego o długości fali 550 nm. Wyniki obliczeń przedstawiono w postaci zależności tak zdefiniowanej wielkości  $\alpha_m$  od zmiennej  $V$ .

Uzyskane przebiegi pokazują, że zależnie od charakterystyki detektora (fotopowej lub skotopowej) zastępcze współczynniki ekstynkcji różnią się o ok. 10 % przy zakresie widoczności  $V$  równym 5000 m. Zmiana zaś rodzaju źródła powoduje różnice rzędu 0,5 %. Przy widocznościach małych różnice te są kilkakrotnie mniejsze. Zależność współczynnika  $\alpha_m$  od  $V$  można przedstawić za pomocą prostego wzoru analitycznego, który z wystarczającą dokładnością aproksymuje tę funkcję.

Z wielkością zastępczego współczynnika ekstynkcji związany jest zakres widoczności  $V_z$ , który wynosi  $3/\alpha_z$ . Różnica między nim a  $V$  oznacza różnicę w ocenie widoczności występującą między wynikami uzyskanymi przy wykorzystaniu światła białego i światła monochromatycznego o długości fali 550 nm. Przy zakresie widoczności 5000 m wynik uzyskany na podstawie pomiaru transmisji światła monochromatycznego o tej długości fali będzie mniejszy o ok. 2 % od oceny wzrokowej dziennej lub otrzymanej przy pomiarze transmisji światła białego zgodnym z definicją MOR. Wzajemne różnice między tymi drugimi wynikami wynoszą ok. 0,5 %. Zależność wielkości różnic od  $V$  można przedstawić za pomocą wzoru przybliżającego, co przy wykorzystaniu podanych w pracy wykresów pozwala na wprowadzenie poprawek redukujących rozbieżności do minimum.

Dodatkowe różnice w ocenie zakresu widoczności powstają w przypadku zastosowania światła monochromatycznego, którego długość fali różni się od 550 nm. Odpowiednią korektę wyniku można przeprowadzić przy wykorzystaniu zależności opisującej zmianę współczynnika ekstynkcji od długości fali lub przez zastosowanie podanej w pracy procedury.

Omówione różnice w ocenie zakresu widoczności są mniejsze niż błędy spowodowane niedokładnością pomiaru transmisji światła, które w praktyce sięgają  $\pm 20$  %. Również ocena wzrokowa widoczności zależna od poziomu radiancji tła i cech indywidualnych obserwatora obciążona jest dużymi błędami. Tak więc różnice spowodowane stosowaniem światła monochromatycznego, zwłaszcza gdy dokonuje się odpowiedniej korekty oceny lub gdy długość fali nie odbiega zbyt od 550 nm, stają się drugorzędne. Ważniejsze jest zapewnienie odpowiednio wysokiej dokładności pomiaru transmisji światła. Zastosowanie laserów półprzewodnikowych stwarza takie możliwości, a ponadto ułatwia budowę transmisometrów o korzystnych cechach konstrukcyjnych i eksploatacyjnych.

## Literatura:

- [1] B. Daino et al., Statistical measurement of atmospheric 6328-A attenuation using a sequential laser transmissometer, *Applied Optics*, 15 (4), 996-999 1976.
- [2] S. Gage et al., *Optoelectronics / Fiber-Optics Applications Manual*, McGraw-Hill Book Co., 1981.
- [3] V.A. Gavrilov, *Visibility in the Atmosphere*, Springfield, VA, 1966.



- [4] M. Gazzi et al., Diagnosis of the causes of systematic errors in the transparency measurements and some experimental verifications in Po Valley, *Journal of Atmospheric and Oceanic Technology*, 2 (6), 201-211 1985.
- [5] D.J. Griggs et. al., The first WMO intercomparison of visibility measurements, *Instruments and Observing Methods*, WMO Report No. 41, 1989.
- [6] ICAO, Meteorological service for international air navigation, Annex 3 to the Convention on International Civil Aviation, (1986).
- [7] C.E. Junge, *Air Chemistry and Radioactivity*, Academic Press, 1963.
- [8] P.W. Kruse et al., *Elements of Infrared Technology: Generation, Transmission and Detection*, J. Wiley a. Sons Inc., 1962.
- [9] *The Photonics Design & Applications Handbook*, Book 3, Laurin Publ.Co. 1991.
- [10] W.E.K. Middleton, *Vision Through the Atmosphere*, University of Toronto Press, 1952.
- [11] R. Synak, Laser transmissometer, *Proc. SPIE , Laser Technology IV*, (w druku).
- [12] R. Synak, Analiza błędów pomiaru transmisji światła, *Oprac. wewn. IMM*, B8, 14-22, 1992.
- [13] D.P. Woodman, Limitations in using atmospheric models for laser transmission estimates, *Applied Optics*, 13 (10), 2193-2195 ,1974.
- [14] E. Zuev, *Laser-light transmission through the atmosphere*, Laser Monitoring of the Atmosphere, Springer Verlag, 1976.



MAREK KOTOWSKI

INSTYTUT MASZYN MATEMATYCZNYCH WARSZAWA

## Programowanie zegara w MS-Windows Programming of the clock in MS-Windows environment

### Streszczenie

W artykule opisano programowanie zegara w środowisku MS-Windows. Przedstawiono praktyczne przykłady zastosowania tej metody do pobierania próbek z analogowo-cyfrowych kart pomiarowych. Opisano podstawowe funkcje związane z realizacją pomiarów i zakodowane w języku Borland C 3.1. Przedyskutowano zalety i wady metody.

### Abstract

In the paper the programming of the clock in MS-Windows environment is described. The examples of practical application of that method are shown for the samples from analog-digital measurements' cards. The basic functions connected with measurements' realization and coded in the Borland C 3.1 language are described. The advantages and disadvantages of the method are described.

### 1. Zegar w MS-WINDOWS

Środowisko MS-WINDOWS, przy wszystkich swoich zaletach, daje użytkownikowi dość ograniczone możliwości korzystania z zegara systemowego. Sprowadzają się one do udostępniania programowi aplikacyjnemu tzw. "wejścia zegarowego", tj. cyklicznego przekazywania programowi sterowania z określoną częstotliwością. Można to zrealizować na dwa sposoby: WINDOWS mogą umieszczać cyklicznie komunikat WM\_TIMER w kolejce związanej z daną aplikacją lub też mogą - również cyklicznie - wywoływać funkcję zdefiniowaną przez użytkownika. Dzięki temu program aplikacyjny może otrzymywać sterowanie co pewien określony interwał czasowy.

Aplikacja inicjuje zegar za pomocą funkcji SetTimer. W jej wywołaniu podaje się identyfikator okna, numer identyfikacyjny inicjowanego zegara, okres - w milisekundach - między zdarzeniami oraz adres funkcji obsługującej komunikaty zegara lub wskaźnik pustą, np.:

```
SetTimer(hWnd, MOJ_ZEGAR, 5000, NULL);
```

Takie wywołanie zainicjuje zegar o numerze określonym przez stałą MOJ\_ZEGAR i spowoduje, że do kolejki komunikatów związanej z oknem o numerze

hWnd co 5000 ms będzie wstawiany komunikat WM\_TIMER. Jeśli czwartym argumentem funkcji nie byłby wskaźnik NULL, musiałby to być adres funkcji, która otrzymywałaby sterowanie co 5 s (opis funkcji SetTimer podany jest w pozycji [2] literatury).

Rozwiązanie to ma jednak wady. Przede wszystkim maksymalna częstota, z jaką program aplikacyjny może otrzymywać komunikaty WM\_TIMER, wynosi 18,2 Hz (patrz niżej). Inaczej mówiąc: komunikaty WM\_TIMER mogą przychodzić nie częściej niż co 55 ms. MS-WINDOWS wprawdzie przyjmie w wywołaniu funkcji SetTimer mniejszą wartość interwału czasowego, np. 10 ms (100 Hz), ale nie uwzględni jej (komunikaty będą wysyłane co 55 ms). Ponadto podana częstota będzie faktycznie zaokrąglana tak, by była wielokrotnością częstoty 18,2 Hz (czas między generowaniem komunikatów WM\_TIMER będzie zaokrąglany do wielokrotności 55 ms). Niezależnie od tego nie ma gwarancji, że program aplikacyjny będzie otrzymywał sterowanie z podaną częstotą. Wystarczy, by system wszedł w tzw. pętlę modalną (np. przy przesuwaniu okna po ekranie) i przez czas jej trwania komunikaty WM\_TIMER nie będą płynąć do programu aplikacyjnego.

Istnieją aplikacje, które wymagają dysponowania mierzalnymi odcinkami czasu krótszymi niż 55 ms, tzn. częstota otrzymywania przez program sterowania musi być większa niż 18,2 Hz. Jedynym możliwym środkiem do wykorzystania, przynajmniej w standardowych komputerach klasy IBM PC, jest tu zegar. Pojawia się problem: jak uzyskać w środowisku MS-WINDOWS przerwanie zegara o większej częstotą? Opiszemy, jak to zrobić.

Najpierw przypomnimy zasady działania zegara w komputerach klasy IBM PC oraz przejmowania i obsługi przerwania w systemie MS-DOS (przez przerwanie będziemy rozumieli tu przerwanie sprzętowe).

## 2. Ogólny opis zegara w IBM PC

Układ programowanego zegara 8253 (*Programmable Interval Timer*) zawiera w sobie trzy niezależne zegary. Cały układ jest sterowany przebiegiem o częstotą 1193181 Hz. Każdy z trzech zegarów może używać tego przebiegu jako podstawy do mierzenia interwałów czasowych, które są całkowitą wielokrotnością jednego cyklu - czasu między dwoma impulsami wejściowymi, czyli 838 ns - bądź do generowania innego sygnału o częstotą będącej ułamkiem częstoty wejściowej. Jeden z zegarów 8253 jest używany do sterowania dynamicznym odświeżaniem pamięci, drugi - do generowania dźwięków. Zegar trzeci używany jest przez BIOS do mierzenia upływu czasu. Generuje on 18,2 razy na sekundę, czyli co 55 ms, przerwanie 8. Program BIOS-u obsługi tego przerwania przy każdym przerwaniu powiększa o 1 systemowy 4-bajtowy licznik impulsów zegara przechowywany w polu pamięci 40:6C. Pole to zawiera liczbę impulsów naliczonych od północy lub od włączenia komputera. Przeliczone na godziny, minuty i sekundy dają aktualny czas ( $1\text{ s} = 18,2\text{ impulsów}$ ,  $1\text{ minuta} = 1092\text{ impulsy}$ ,  $1\text{ h} = 65543\text{ impulsy}$ ). Oprócz tego program obsługi przerwania 8 zmniejsza wartość licznika czasu włączenia silnika napędzającego dysk miękki (pole pamięci 40:40). Gdy wartość tego licznika stanie się równa 0, silnik jest wyłączany.

Liczba 18,2 nie jest przypadkowa. Zegar generujący przerwanie 8 programuje się podając mu wartość, przez którą należy podzielić częstota 1193181. Wartością tą jest inicjowany 16-bitowy licznik, który jest zmniejszany o 1 przy każdym impulsie



wejściowym - czyli co 838 ns - i kiedy osiągnie wartość 0, generowane jest przerwanie 8. BIOS programując zegar jako wartość dzielnika podaje 0; a ponieważ zawartość licznika najpierw jest zmniejszana o 1, a dopiero potem jest sprawdzane, czy jest ona równa 0, zatem podanie 0 jest równoważne podaniu 65536 (przy pierwszym impulsie wejściowym odjęcie 1 od 16-bitowego rejestru zawierającego 0 nada mu wartość 65535). Z kolei  $1193181/65536 = 18,20651 \approx 18,2$ .

Wynika stąd, że 18,2 Hz jest najmniejszą możliwą częstotnością przerwań zegara. Można ją zwiększyć (jak to zrobić, opisano w dalszej części rozdziału), ale nie można jej zmniejszyć.

Adresy programów obsługi przerwań znajdują się w tzw. tablicy wektorów przerwań (w systemie DOS zajmuje ona pierwszych 1024 bajtów pamięci). Tablicę tę wypełnia BIOS i DOS (modyfikuje ją także MS-WINDOWS). Aby przejąć przerwanie, tj. zainstalować w systemie program jego obsługi, trzeba w odpowiednie miejsce tablicy wstawić adres tego programu.

Przejęcie przerwania nie jest równoznaczne z jego obsługą. Ponieważ nie wiadomo, jakie programy przejęły wcześniej to przerwanie i jak go wykorzystują, trzeba wywoływać pierwotny program obsługi tego przerwania (o ile wektor danego przerwania nie był pusty). Program przejmujący przerwanie zegara może wykonywać różne operacje, np. testować stany poszczególnych urządzeń (klawiatura, RS-232C itp.). Można przy tym, przez odpowiednie zaprogramowanie układu 8253, uzyskać znacznie częstsze przerwania zegara (patrz p. 4.1.1), a tym samym i możliwość odpowiednio częstszego badania w systemie wybranych stanów. Niekiedy zmiana częstotliwości przerwań jest wręcz niezbędna. Częstota 18,2 Hz (wartość tę nazywać będziemy dalej standardową) wystarcza np. do sprawdzania stanu klawiatury, kiedy chce się, by program przejmujący przerwanie zegara reagował na naciśnięcie pewnych klawiszy stanu (CTRL, ALT, SHIFT), ale jest już za mała do badania np. stanu aparatury pomiarowej. Innym przykładem sytuacji wymagającej zwiększenia częstotliwości przerwań zegara jest generowanie dźwięków o czasach trwania krótszych niż 1/18 s i niezależnych od szybkości procesora.

Mimo zwiększonej częstotliwości przerwań zegar systemu musi pracować tak samo. Oznacza to m.in., że 1 impuls musi być dodawany do licznika impulsów co osiemnastą część sekundy, niezależnie od tego, jak częste będą przerwania zegara. Stąd wniosek: w zainstalowanym programie obsługi tego przerwania, chociaż będzie on wywoływany częściej niż 18,2 raza na sekundę, jako że częściej będzie generowane przerwanie zegara, nie za każdym razem będzie wywoływany pierwotny program obsługi przerwania 8. Będzie on wywoływany 18,2 raza na sekundę (przerwania, przy których nie będzie on wywoływany, nazwiemy przerwaniem nadmiarowym). Zatem jakkolwiek przerwanie zegarowe będzie generowane częściej, z punktu widzenia zegara systemowego wszystko będzie bez zmian.

Zalóżmy, że chcemy zwiększyć częstotliwość przerwań zegara 64 razy. Zwiększyć częstotliwość przerwań znaczy zmniejszyć wartość dzielnika programującego układ 8253, czyli wysłać do odpowiedniego portu nie 0 (jest to równoważne 65536), ale  $65536/64 = 1024 = 0x400$ . Zatem w czasie jednej sekundy będzie generowanych  $64 \cdot 18,2 = 1165$  przerwań. Należy wprowadzić w programie obsługi przerwania 8 licznik przerwań nadmiarowych i co 64 takie przerwania wywoływać pierwotny program obsługi przerwania 8. Ilustruje to rys. 1. Duża litera X oznacza przerwanie, przy którym jest

zwiększana liczba naliczanych impulsów (wywoływany jest pierwotny program obsługi przerwania nr 8), a mała litera x - przerwanie nadmiarowe.

Przerwania o częstości standardowej (18,2)  
-----X-----X-----

Przerwania o częstości zwiększonej 64 razy  
x-x-x-x-X-x-x-.....x-x-X-x-x-x-x-x-x-

64 razy

Rys.1. Zmiana częstości przerwań zegara

Ogólny wzór na wartość licznika L, jaką trzeba programować zegar, by uzyskać przerwania o częstości C (zakładamy, że C jest nie mniejsze niż 18,2065), ma postać

$$L = \frac{65536}{C} \times 18,2065$$

Oprócz tego trzeba wiedzieć, co ile przerwań należy wywołać pierwotny program obsługi przerwania zegara. Wartość tę oblicza się według wzoru:

$$N = \frac{C}{18,2065}$$

Dla przykładu, jeśli nowa częstość przerwań zegara ma wynosić 700 Hz, to wartość licznika wyniesie

$$L = \frac{65536}{700} \times 18,2065 = 1704,544 \approx 1705$$

zaś liczba przerwań nadmiarowych

$$N = \frac{700}{18,2065} = 38,44 \approx 38$$

I ostatnia bardzo istotna uwaga: pierwotny program obsługi przerwania zegara wysyła polecenie EOI (*End Of Interrupt*) do PIC (*Programmable Interrupt Controller*). Jeśli przy przerwaniu zegara nie wywołuje się tego programu (przerwanie nadmiarowe), program przejmujący przerwanie 8 musi sam wysłać takie polecenie. Innymi słowy: musi obsłużyć przerwanie nadmiarowe. Trzeba o tym pamiętać, w przeciwnym bowiem razie dane przerwanie będzie traktowane jako nieobsłużone i system się zawiesi.

## 2.1. Biblioteczne funkcje przejmowania przerwań .

Do przejmowania przerwań w języku C można wykorzystać funkcje biblioteczne, które istnieją w kompilatorze BORLAND C 3.1 (zazwyczaj istnieją one w bibliotekach innych kompilatorów języka C dla komputerów klasy IBM PC). Oto ich prototypy:



```
void interrupt(*getvect(unsigned NumerPrzerw)) ();
void setvect(int NumerPrzerw,
             void interrupt (*Funkcja) ( ));
```

Pierwsza funkcja pobiera adres funkcji obsługi przerwania o podanym numerze. Druga instaluje nowy program obsługi przerwania.

Atrybut **interrupt** odnosi się do funkcji, która ma przejmować przerwania. Powoduje on, że kompilator generując kod wynikowy tej funkcji doda na jego początku instrukcje przechowujące zawartości wszystkich rejestrów, a na końcu funkcji wstawi sekwencję odtwarzającą je. Ponadto sam powrót z funkcji będzie realizowany za pomocą instrukcji maszynowej **IRET**, a nie **RET**.

**UWAGA:** Trzeba pamiętać, że sama realizacja przerwania, a tym bardziej jego obsługa, zajmują czas procesora. Przy dużym zwiększeniu liczby przerwania lub przy wolnym komputerze, może nastąpić wyraźne spowolnienie wykonywania programów aplikacyjnych, czy nawet poleceń wewnętrznych DOS. W krańcowym przypadku może to spowodować zawieszenie się systemu.

### 3. Przeprogramowanie zegara

Przeprogramowanie zegara sprowadza się do przejścia przerwania 8 i zmiany częstości zegara na większą.

Po to by zaprogramować zegar na nową częstość przerwania, należy (dla uproszczenia podamy same wartości wysyłane do portów):

- wysłać do portu 0x43 wartość 0x36,
- wysłać do portu 0x40 wartość licznika programującego.

Programowanie częstości zegara powinno być realizowane przy maskowanych przerwaniach, aby uniknąć interferencji między programowaniem zegara a przerwaniem 8 (przedstawiono to w pkt 4.1).

Obsługa przerwania tak przeprogramowanego zegara musi być oczywiście realizowana przez odpowiednią funkcję obsługi, której adres musi znajdować się w tablicy wektorów przerwania, w miejscu odpowiadającym przerwaniu 8 (adres 0x20:0).

### 4. Opis przykładowej aplikacji w MS-WINDOWS - system pomiarowy VIRT

Przeprogramowanie zegara w środowisku MS-WINDOWS zilustrujemy na przykładzie zaprojektowanego w IMM oprogramowania systemu pomiarowego VIRT II, który został napisany w języku BORLAND C 3.1.

Środowisko MS-WINDOWS wybrano ze względu na przyjazny interfejs użytkownika i łatwość graficznego kreślenia jednocześnie wyników kilku pomiarów. Dane pomiarowe reprezentowane są przez liczby całkowite ze znakiem (zakres wartości od - 32768 do + 32767) i wyświetlane w oknie utworzonym za pomocą standardowych funkcji bibliotecznych BORLAND C 3.1 dla MS-WINDOWS. Ponieważ pomiary realizowane są zazwyczaj z częstością większą od standardowej, system VIRT musiał dysponować mechanizmem obliczania interwału czasowego krótszego niż 55 ms. W takiej sytuacji standardowe środki MS-WINDOWS były niewystarczające. Trzeba było przeprogramować zegar.

Opiszemy podstawowe funkcje związane z programowaniem zegara i obsługą jego przerwai. Podamy je w uproszczonej postaci, ograniczając się tylko do problemów związanych z przeprogramowywaniem zegara i wykorzystaniem go (pominiemy szczegóły związane z wewnętrzną strukturą systemu VIRT). Następnie przedstawimy ogólną zasadę działania systemu.

#### 4.1. Inicjowanie zegara

Wszystkie funkcje związane z obsługą zegara znajdują się w oddzielnym module. Pierwszą z nich jest funkcja **InicjujZegar**, służąca do inicjowania całego mechanizmu. Realizuje ona następujące operacje:

- przechowuje pierwotny wektor przerwania 8,
- przejmuje przerwania zegara,
- ustawia nową częstotliwość przerwai.

Jako argumenty funkcji podawane są:

- nowa częstotliwość przerwai,
- adres funkcji pomiarowej,
- adres zmiennej `Status`, służącej do kontrolowania, czy przerwania nie nalożyły się.

Zdefiniujemy dwie makroinstrukcje maskowania i dopuszczania przerwai:

```
#define MASKUJ asm{ \
                CLI \
                }
```

```
#define DOPUSC asm{ \
                STI \
                }
```

Oto funkcja **InicjujZegar**:

```
int InicjujZegar(unsigned int Czestosc,
                (*) (Func),
                unsigned int *StatusZegara)
{
    /* POBRANIE WSKAZNIKA NA FUNKCJE PIERWOTNA */
    Pierwotna=getvect(8);

    /* MASKOWANIE PRZERWAŃ ZEGARA */
    MASKUJ;

    /* PRZEPROGRAMOWANIE ZEGARA */
    ProgramujZegar(Czestosc);

    /* USTAWIENIE NOWEGO WEKTORA PRZERWANIA */
    setvect(8, Obsluga8H);
}
```



```

Status=StatusZegara;
*Status=0;
/* DOPUSZCZENIE PRZERWAŃ ZEGARA */
DOPUSC;
return(1);
}

```

Funkcja **ProgramujZegar** oraz funkcja obsługi przerwania zegara **Obsluga8II** opisane są niżej. Zmienna **Pierwotna** jest jedną ze zmiennych prywatnych (tj. klasy **static**) w module zegara. Oto opis dwóch pierwszych zmiennych (dodatkowo zostaną podane niżej):

```
interrupt (*Pierwotna)();
```

Adres pierwotnego programu obsługi przerwania 8.

```
int *Status;
```

Adres zmiennej w module wywołującym, która informuje, czy nastąpiło nałożenie się przerwania.

#### 4.1.1. Programowanie zegara

Zanim opiszemy funkcję przeprogramowania zegara, przedstawimy kolejne dwie zmienne prywatne modułu zegara:

```
int PrzerwaniaNadm;
```

Liczba przerwania nadmiarowych, przy jakich program pierwotny nie jest wywoływany (jest ona stała dla danej częstotliwości).

```
int LicznikPrzerwan;
```

Licznik przerwania nadmiarowych, dekrementowany przy każdym przerwaniu (gdy jego wartość stanie się równa 0, wywoływany jest pierwotny program obsługi przerwania zegara).

Oto funkcja przeprogramowania zegara. Jej argumentem jest nowa częstotliwość przerwania. Zakładamy, że częstotliwość ta jest w zakresie  $18 \div 32767$  Hz. Co istotne: zakładamy tu, że nowe częstotliwości są większe od częstotliwości standardowej.

```
static int ProgramujZegar(int Czestosc)
```

```
{
```

```
float y;
```

```
int WartoscDzielnika;
```

```
if (Czestosc < 19)
```

```
return(0);
```

```
y=(float)Czestosc/18.2065;
```

```
/* INICJOWANIE LICZNIKOW */
```

```
LicznikPrzerwan=(int)y;
```

```
PrzerwaniaNadmiarowe=(int)y;
```

```

/* TERAZ USTAWIENIE LICZNIKA */
WartoscDzielnika=(int) (65536.0/y);
outportb(0x43, 0x36);
outportb(0x40, WartoscDzielnika%256);
outportb(0x40, WartoscDzielnika/256);

return (1);
}

```

Jest to wersja uproszczona, zakładająca, że nowa częstość przerwań jest wyższa od częstości standardowej. Częstości mniejsze można obsługiwać za pomocą komunikatów WM\_TIMER. Można to również zrobić w procedurze obsługi przerwania zegara i tak zrobiono w systemie VIRT (patrz 5.3.).

## 4.2. Inicjowanie pomiaru

Bezpośrednio po wykonaniu funkcji **InicjujZegar** przerwania zegara pojawiają się z nową częstością, ale pomiary nie są przeprowadzane. Uruchamia je dopiero funkcja **InicjujPomiar**, inicjując odpowiednie zmienne prywatne modułu (funkcja obsługi przerwania uwzględnia już zwiększoną liczbę przerwań). Sam pomiar fizyczny realizowany jest przez inną funkcję (patrz 4.2.2.).

### 4.2.1. Bufor okrężny

Zanim przedstawimy funkcję inicjującą pomiar, opiszemy bufor okrężny (*circular buffer*). Wykorzystywany on jest do zapisywania wyników pomiarów, które później poddawane są obróbce i obrazowane na ekranie. Oto struktura, która go opisuje.

```

typedef struct
{
    int *Bufor;                /* ADRES POZATKU BUFORA
                               ZAWIERAJACEGO POMIARY */
    int RozmiarBufora;         /* ROZMIAR BUFORA */
    int WskZapisu;             /* WSKAZNIK MIEJSCA, GDZIE BEDZIE
                               ZAPISANA NASTEPNA PROBKKA */
    int WskOdczytu;            /* WSKAZNIK MIEJSCA, SKAD ZOSTANIE
                               ODCZYTANA NASTEPNA PROBKKA */
    unsigned int Status; /* STATUS 1 - OK!, 0 -
                               PRZEPelnNIENIE BUFORA */
} BUFOR_POMIAROW;

```

Przy każdym zapisie pomiaru do bufora wskaźnik zapisu zwiększa się o 1, zaś przy odczycie pomiaru z bufora zwiększa się z kolei o 1 wskaźnik odczytu. Jeśli wskaźnik zapisu "dogoni" wskaźnik odczytu, bufor jest zapełniony i dalsze wpisywanie do niego próbek staje się niemożliwe.



#### 4.2.2. Funkcja inicjująca pomiar

Funkcja inicjująca pomiar ustawia dwie zmienne prywatne modułu zegara: adres funkcji realizującej pomiar i adres bufora okrężnego, po czym ustawia znacznik, że pomiar ma być realizowany. Oto odpowiednie trzy zmienne prywatne modułu zegara:

```
void interrupt (*FPom) ();
```

Adres funkcji realizującej pomiar.

```
BUFOR_POMIAROW *BuforOkreznny;
```

Adres bufora okrężnego.

```
int Pomiar;
```

Informuje, czy pomiary mają być realizowane. Początkowo wartość tej zmiennej wynosi 0 (pomiary nie są realizowane).

A oto sama funkcja inicjująca pomiar:

```
int InicjujPomiar((*) (Func),
                  BUFOR_POMIAROW *Bufor)
{
    FPom=Func;
    BuforOkreznny=Bufor;
    Pomiar=1;
}
```

Bezpośrednio po nadaniu wartości 1 zmiennej Pomiar rozpoczyna się realizowanie pomiarów.

#### 4.3. Obsługa przerwania zegara

Przedstawimy teraz nową funkcję obsługi przerwania zegara. Jej adres został wstawiony do tablicy wektorów przerwania przez funkcję InicjujZegar.

```
void interrupt Obsluga8H(...)
```

```
{
    static int Blokada=0;

    if (--LicznikPrzerwań == 0)
    {
        /* WYWOŁANIE PIERWOTNEJ FUNKCJI OBSŁUGI */
        (*Pierwotna)(...);
        LicznikPrzerwań = PrzerwaniaNadmiarowe;
    }
    else
        outportb(0x20, 0x20); /* "END OF INTERRUPT" */

    /* TO, CO WYZEJ, JEST ZWIĄZANE TYLKO Z OBSŁUGĄ PRZERWAŃ
       O NOWEJ CZĘSTOŚCI. TERAZ NASTĘPUJE FRAGMENT
       REALIZUJĄCY POMIARY */
}
```

```

if (Pomiar) /*CZY ZAINICJOWANO POMIARY? */
/* TERAZ NASTĘPUJE REALIZACJA POMIARÓW */
if (Blokada == 0) /*CZY NIE JEST POMIAR W TRAKCIE?*/
{
    Blokada=1;
    if (ZrobPomiar() == 0) /* CZY BUFOR ZAPEŁNIŁ SIĘ?*/
        BuforOkreznny->Status=0;
    Blokada=0;
}
else /* PRZERWANIA NAKŁADAJĄ SIĘ */
    (*Status)++;
}

```

Znacznik **Blokada** został wprowadzony dla celów ochrony przed nakładaniem się przerwań. Jeśli pomiar trwałby zbyt długo i w trakcie jego realizacji pojawiłoby się nowe przerwanie zegara, nieczłowa wartość znacznika **Blokada** uniemożliwi ponowne wywołanie funkcji **ZrobPomiar**, zaś wartość zmiennej **Status** zostanie zwiększona o 1 (informuje ona, że przerwania nałożyły się, umożliwiając podjęcie stosownych kroków przez program aplikacyjny).

#### 4.4. Zapisywanie pomiaru

Funkcja **ZrobPomiar**, wywoływana w funkcji **Obsługa8II**, realizuje pomiar i wpisuje otrzymaną wartość do bufora okrężnego (została ona również uproszczona). Funkcja zwraca wartość 1, jeśli pomiar został wpisany do bufora, i 0 jeśli bufor został zapełniony. Sam pomiar fizyczny realizowany jest przez funkcję, której adres został podany w wywołaniu funkcji **InicjujPomiar** (został on przechowany w zmiennej prywatnej **FPom**):

```

int ZrobPomiar(void)
{
    int k;
    int WartPomiar;

    BUFOR_POMIAROW *p=BuforOkreznny;

    k=p->WskOdczytu - p->WskZapisu;

    /* CZY BUFOR JEST PEŁNY? */
    if ((k == 1) || (k == - (p->RozmiarBufora - 1)))
        return(0);

    /* WYKONANIE POMIARU I ZAPISANIE WYNIKU POMIARU DO
       BUFORA */
    *(p->Bufor + p->WskZapisu)=(*FPom)();

    /* ZWIEKSZENIE WSKAZNIKA ZAPISU */
    if(p->WskZapisu == p->RozmiarBufora - 1)

```



```

p->WskZapisu=0;
else
p->WskZapisu++;

return(1);}

```

Pomiary realizowane przez funkcję **ZrobPomiar** są wpisywane do bufora okrężnego i wskaźnik zapisu jest zwiększany. Jeśli wskaźnik odczytu bufora okrężnego nie byłby zmieniany (patrz 4.5), bufor zostałby zapelniony i - jak widać w przedstawionej wyżej funkcji - następne pomiary nie byłyby realizowane.

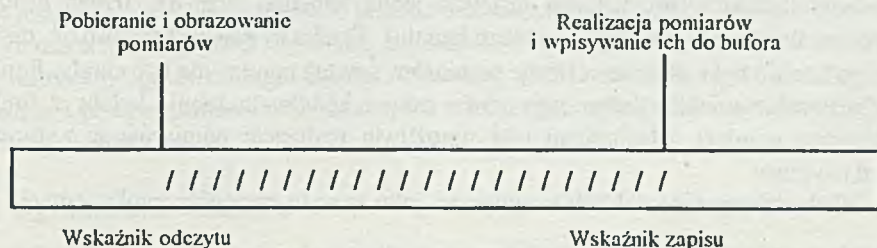
#### 4.5. Obrazowanie pomiarów

W funkcji obsługi przerwania zegarowego dane pomiarowe są jedynie zapisywane do bufora okrężnego. Żadne inne operacje, zwłaszcza te, które wymagają wywołania funkcji systemowych, nie są na nich realizowane. Wynika to z asynchroniczności przerwania zegara - próba wywołania funkcji systemowej w funkcji obsługi przerwania zegara mogłaby doprowadzić do interferencji między funkcjami systemowymi.

System VIRT II, przejmując przerwanie 8 i zmieniając częstość przerwania zegara, jednocześnie inicjuje zegar systemowy za pomocą opisanej na wstępie funkcji **SetTimer**, określając standardowy odcinek czasowy między komunikatami **WM\_TIMER** (55 ms). Tym samym odpowiedni fragment kodu VIRT otrzymuje sterowanie 18,2 raza na sekundę (faktycznie komunikaty **WM\_TIMER** generowane są przy wywołaniu pierwotnego programu obsługi zegara w procedurze **Obsługa8II**). Przy każdym kolejnym pojawieniu się komunikatu **WM\_TIMER** bufor okrężny zawiera pomiary wykonane w czasie od pojawienia się poprzedniego komunikatu **WM\_TIMER**. Odpowiednia funkcja wyświetla wyniki tych pomiarów w oknie, zwiększa wskaźnik odczytu i oddaje sterowanie. Następne pomiary, w tym te zapisane do bufora okrężnego w trakcie wyświetlania pomiarów poprzednich, zostaną wyświetlone przy następnym komunikacie **WM\_TIMER**. Obrazuje to rys. 2:

**Moduł z obsługą komunikatów wejściowych WM\_TIMER :**

**Moduł funkcji zegara**



Rys 2. Wypełnianie bufora okrężnego i pobieranie z niego danych.

#### 4.6. Pomiary współbieżne

Dotychczas była mowa o realizowaniu jednego pomiaru. Jest możliwe również realizowanie równolegle kilku procesów pomiarowych w trakcie jednego przzerwania zegarowego. Załóżmy, że w karcie analogowo-cyfrowej jest 16 kanałów i że próbki z każdego z nich mają być odczytywane i zapisywane niezależnie. Każdemu kanałowi zatem musi być przyporządkowana oddzielna struktura BUFOR\_POMIAROW. Można założyć, że pomiary są realizowane z taką samą częstotnością i przez tę samą funkcję. Ale jest to uproszczenie; może być też tak, że pomiar w każdym kanale realizuje inna funkcja i że częstotść pomiarów w każdym kanale jest inna (musi ona być przy tym całkowitym podzielnikiem częstotści przerwań). Można również wprowadzić liczbę pomiarów, po wykonaniu których próbkowanie w danym kanale jest przerywane.

Po to aby realizować pomiar we wszystkich kanałach, należy wstawić do funkcji ObsługaII zamiast funkcji ZrobPomiar pętlę:

```
for (k=0; k < 16; k++)
    if (ZrobPomiar(k) == 0)
        return (0);
```

gdzie k oznacza numer kanału. Nicco bardziej skomplikowany byłby również opis parametrów związanych z kartą. Wygodnie jest zdefiniować strukturę opisującą kanał:

```
typedef struct
{
    unsigned int Pomiar;      /* STAN KANAŁU */
    unsigned int LiczbaPom;  /* LICZBA POMIAROW */
    (*Func) (int);           /* ADRES FUNKCJI POMIAROWEJ */
    int Czestosc;            /* CZESTOSC POMIAROW */
    BUFOR_POMIAROW *Bufor;  /* ADRES BUFORA POMIAROWEGO */
} KANAŁ;
```

Kanały karty analogowo-cyfrowej można opisać za pomocą tablicy:

```
KANAŁ Kanały[16];
```

W tej sytuacji odpowiednie zmienne prywatne modułu zostałyby zastąpione przez składowe struktury (np. zamiast testować jedną zmienną Pomiar, trzeba testować ladowe Pomiar w strukturze każdego kanału). Trzeba by również sprawdzić, czy dla danego kanału była ustawiona liczba pomiarów, czy też pomiar ma być ciągły. Funkcja InicjujPomiar miałaby jeden argument - numer kanału (podobnie byłoby z funkcją realizującą pomiar). Mechanizm taki umożliwiałby realizację pomiarów w wybranych kanałach karty.

Nie będziemy tego dokładnie opisywać, jako że są to szczegóły implementacji.

#### 4.7. Inne funkcje obsługi zegara w systemie VIRT

Oprócz opisanych funkcji w module zegara są również inne funkcje publiczne, które okazały się przydatne, czy wręcz niezbędne. Oto one:

```
void BlokujPomiar(int Kanał);
Blokuje pomiar w kanale.
```



```
void ZwolnijPomiar(int Kanal);
```

Wznawia realizację pomiarów.

```
int ZmienCzestość(int Czestosc);
```

Zmienia częstość przerwań zegara.

```
void ZamknijPomiar(int Kanal);
```

Kończy próbkowanie w danym kanale.

```
void KoniecSesji(void);
```

Kończy próbkowanie i przeprogramowuje zegar na częstość standardową (odtwarza również pierwotny wektor przerwania zegara).

## 4.8. Parametry implementacji

System VIRT II uruchomiono na komputerze IBM PC/386 z zegarem 40 MHz. System pracował z kartą przetwornika a/c i c/a AD611B w trybie programowej obsługi i był w stanie pobierać 1600 próbek z 16 kanałów równoległe, czyli łącznie 25600 próbek na sekundę (przetwornik pracował w trybie pomiaru w 16 kanałach). Pomiaru wpisywane były do buforów okrężnych (każdy kanał miał niezależny bufor), ale nie obrazowane na ekranie; po zapelnieniu się buforów ich zawartość była przepisywana do buforów pośrednich i dopiero po zakończeniu pomiaru obrazowana na ekranie.

Czas, jaki pochłania wizualizacja wyników pomiarów, zależy od kilku czynników, w tym m.in. od liczby pomiarów, od tego czy oprócz wizualizacji dane są poddawane także pewnej obróbce, czy próbki z poszczególnych kanałów są obrazowane w jednym oknie, czy też każdemu kanałowi przypisane jest inne okno itd. Istotny tu jest też rodzaj karty pomiarowej i rodzaj sterownika monitora ekranowego.

Dla przykładu, zobrazowanie 400 próbek ze wszystkich 16 kanałów (łącznie 6400 próbek), wyświetlanych w jednym oknie, zajmującym cały ekran, trwało 1,5 s (próbki nie były poddawane dodatkowej obróbce).

## 5. Wady i zalety rozwiązania

W prezentowanej metodzie zastosowano rozwiązanie niskopoziomowe. W dokumentacji MS-WINDOWS nie ma zastrzeżenia, że nie wolno przeprogramowywać zegara i nie jest to zakazane, można zatem uznać metodę za całkowicie poprawną, tym bardziej jeśli daje poprawne wyniki. Z punktu widzenia MS-WINDOWS cały mechanizm jest przezroczysty; pojawił się tylko pośrednik między przerwaniami sprzętowym a pierwotnym programem jego obsługi. Niemniej jednak nie o tę należy stosować ostrożnie i przeprogramowywać zegar tylko wtedy, gdy mają być realizowane operacje, które tego wymagają.

### 5.1. Ograniczenia czasowe

Większość problemów związanych z omawianym rozwiązaniem dotyczy ograniczeń czasowych.

Przed wszystkim ogólny czas obsługi przerwania musi być możliwie krótki. Warunkiem koniecznym dokonywania pomiarów jest, by nie nastąpiło nałożenie się przerwań. Wprawdzie można w funkcji *Obsługa8II* nie dopuszczać następnych przerwań, nie wykonując instrukcji *STI*, ale problem i tak pojawia się przy pierwszym wy-

wołaniu pierwotnego programu obsługi przerwania 8, który dopuszcza przerwanie. Jeśli pomiary są jednorodne, tzn. każdy pomiar zabiera podobną ilość czasu (tak jest w przypadku systemu VIRT II), nałożenie się przerwania oznacza praktycznie blokadę pracy i powinno powodować kończenie procesu pomiarowego. Jeśli przy pomiarach istnieją pewne stany specjalne, których obsługa może zabrać więcej czasu, można rozważyć dopuszczenie nałożenia się przerwania przy obsłudze takich stanów (może to się wiązać z utratą jednej lub więcej próbek pomiarowych).

W celu zminimalizowania czasu obsługi przerwania zegarowego można:

- kompilować funkcję obsługi przerwania i funkcje pomocnicze przez nią wywoływane w modelach małego programu i małych danych, co zmniejszy czas wywoływania funkcji i przekazywania argumentów (wyjątkiem są argumenty związane z obsługą bufora okrężnego, w których może występować daleki wskaźnik),
- kodować funkcje pomocnicze w konwencjach skracających czas ich wywoływania i wykonania (np. *register* w nomenklaturze kompilatora BORLAND C 3.1),
- zakodować wszystkie funkcje obsługi zegara w języku assemblera,
- wybrać szybszy komputer.

Gdy częstość pomiarów musi być wysoka, wyjściem krańcowym i drastycznym jest zrezygnować na czas dokonywania pomiarów z wywoływania pierwotnego programu obsługi przerwania zegara. Rozwiązanie takie, jakkolwiek z punktu widzenia realizacji pomiarów i ich efektywności jest korzystne, ma jednak ujemne skutki, w szczególności:

- mysz praktycznie będzie nieaktywna,
- komunikaty WM\_TIMER nie będą generowane (można będzie zapisać tylko tyle próbek, ile zmieści się w buforze okrężnym i ewentualnie w buforach pomocniczych),
- czas systemowy ulegnie zawieszeniu (można będzie uaktualnić go po zakończeniu serii pomiarów, wiedząc ile czasu trwały),
- wstrzymanie procesu pomiarowego musiałoby być realizowane przez inne przetwarzanie, np. przerwanie klawiatury.

Niezależnie od problemu ograniczeń czasowych związanych z obsługą przerwania istnieje problem ograniczeń czasowych związanych z ilustrowaniem pomiarów. Jeśli ma ono być realizowane na bieżąco za pomocą środków MS-WINDOWS, może okazać się, że procedury WINDOWS nie będą nadążać z wyświetlaniem wyników (opóźnienie będzie występować przede wszystkim przy przesuwaniu zawartości okna), wskaźnik odczytu bufora okrężnego nie będzie dostatecznie szybko zmieniany i bufor okrężny zapelni się.

### 5.1.1. Ograniczenia na częstość pomiarów

Częstość pobierania próbek z zasady nie może być większa od częstości przerwania zegara. Jeśli częstość przerwania jest częstością pobierania próbek - problemu nie ma. Nicco gorzej jest, gdy jest ona mniejsza. Jeśli częstość przerwania wynosi 1000 Hz, a częstość pobierania próbek jest dwa razy mniejsza, pomiar realizowany jest co drugie przerwanie. Podobnie można realizować pomiary z częstościami 50 (pomiar co 20 przerwania), 100 (pomiar co 10 przerwania), 200 (pomiar co 5 przerwania). Wszystkie te



liczby są całkowitymi dzielnikami liczby 1000. Gorzej jest z częstotliwościami nie będącymi dzielnikami liczby 1000, np. 150, 300, 400. W takich sytuacjach można wybrać różne rozwiązania.

Pierwszym jest zaprogramowanie zegara dokładnie na tę częstotliwość zegara, np. 400. W takim przypadku przy każdym przerwaniu realizowany byłby jeden pomiar w jednym z kanałów. Rozwiązanie to jest poprawne i dokładne. Jego ograniczeniem jest to, że można by realizować w zasadzie tylko pomiary z daną częstotliwością lub pomiary o częstotliwości będącej całkowitym dzielnikiem częstotliwości przerwania (np. 50, 100, 200).

Można również, zachowując częstotliwość przerwania, realizować pomiary z częstotliwością przybliżoną (np. dla pomiaru z częstotliwością 300 Hz, przy częstotliwości zegara 1000, realizować pomiar przy co trzecim przerwaniu). Jest to przybliżenie, które w zależności od okoliczności może być akceptowane lub nie.

Oczywiście wszystkie te rozważania dotyczą standardowego zegara IBM PC. Jeśli w danej wersji komputera istnieją inne zegary, można je wykorzystać.

## 5.2. Częstotliwości mniejsze od standardowej

Z założenia pomiary realizowane są z częstotliwościami większymi od standardowej. Częstotliwości mniejsze, praktycznie w tym przypadku niewykorzystywane, mogą jednak być również brane pod uwagę, przynajmniej teoretycznie. Oczywiście można je obsługiwać za pomocą nowego zegara MS-WINDOWS, uzyskanego poprzez odpowiednie wywołanie funkcji `SetTimer`. Ale takie rozwiązanie, wprowadzić poprawne, byłoby niespójne - część próbkowania byłaby realizowana w module obsługi zegara, a część przy obsłudze komunikatu `WM_TIMER`.

Częstotliwości mniejsze od standardowej można również obsłużyć w funkcji `Obsługa8II`. Zachodzi wtedy istotna różnica w stosunku do koncepcji przedstawionej wyżej: przy każdym przerwaniu wywoływany jest pierwotny program obsługi przerwania, ale nie przy każdym jest realizowany pomiar (np. przy częstotliwości 9 pomiar realizowany jest co drugie przerwanie). Funkcja obsługi przerwania komplikuje się nieco, zwłaszcza jeśli w systemie mają być realizowane zarówno pomiary o częstotliwości wyższej, jak i niższej od częstotliwości standardowej, ale za to całość pomiarów realizowana jest w sposób funkcjonalnie jednorodny.

## 6. Inne zastosowania

Opisana metoda może być stosowana we wszelkich aplikacjach realizowanych w środowisku MS-WINDOWS i wymagających interwałów krótszych niż osiemnasta część sekundy. Dotyczy to generowania złożonych dźwięków, tworzenia histogramów itp. Można ją również stosować w innych środowiskach operacyjnych, np. w DOS-ie, ale trzeba pamiętać o asynchroniczności przerwania zegara i konieczności unikania interferencji funkcji systemowych (patrz : [4]).

## Literatura

- [1] Interfejs graficzny użytkownika (VIRT II). Opracowanie wewnętrzne IMM.
- [2] Microsoft Windows, Software Development Kit, Reference 3.0, Microsoft Corporation, 1990.

- [3] Jones McCord - Developing Windows Applications with Borland C 3.1, Second edition, SAMS, 1992.
- [4] M.Kotowski - Pod zegarem, Lupus, Warszawa, 1992 r.



**RAFAL OBUCHOWSKI**

**INSTYTUT KOMPUTEROWYCH SYSTEMÓW  
AUTOMATYKI I POMIARÓW WROCŁAW**

## **Komputerowa sieć sterująca LON Computer control network LON**

### **Streszczenie**

Instytut Komputerowych Systemów Automatyki i Pomiarów uzyskuje licencję LONWORKS Independent Developer (LID) firmy ECHELON Corporation, USA. W artykule opisano komputerową sieć sterującą LON, jej elementy sprzętowe i oprogramowanie.

### **Abstract**

Computer Systems Automatics and Measurement Institute obtains license the LONWORKS Independent Developer (LID). The paper describes an idea of local operating network – LON, it hardware and software.

Obecnie wyraźnie jest wyczuwalna tendencja do rozwoju systemów sterowania i automatyki w kierunku systemów o coraz wyższym stopniu rozproszenia inteligencji. Rozwiązania bazują na coraz tańszych i mniejszych komputerach, które łączy się w sieci tworzące systemy kontroli. Bardzo duży nacisk jest kładziony na rozwój i doskonalenie komunikacji między elementami sieci. Zagadnienie komunikacji staje się najważniejszym elementem nowoczesnych systemów kontrolno-sterujących dnia dzisiejszego.

Dawne scentralizowane systemy sterowania były i są trudne do realizacji, wymagają dużych nakładów przy instalacji, a wszelkie rekonfiguracje nastroczają licznych problemów.

Wysoki poziom technologii półprzewodnikowych umożliwia szerokie zastosowanie sprawniejszych środków technicznych w procesach automatyzacji systemów kontrolno-sterujących o inteligencji rozproszonej, takich jak np.: kontrola i atomatyzacja procesów przemysłowych, kontrola budynków, kontrola i sterowanie maszyn, pojazdów, urządzeń domowych itp.

Istnieje wiele standardów magistral sieci komunikacyjnych umożliwiających realizację rozproszonych systemów sterowania, np.: BitBus (Intel Co.), CAN (Bosch GmbH), MILSTD-1553B (US Air Force), ProfiBus oraz nowe, niezależne rozwiązanie

LONWORKS™ lokalnej sieci sterowania LON (*Local Operating Network*) firmy ECHELON z Palo Alto, California USA.

LONWORKS charakteryzuje się absolutnie nowatorskimi rozwiązaniami na wielu płaszczyznach realizacyjnych. Sieci LON stanowią nową klasę rozwiązania technicznego, które pozwala wprowadzić inteligencję do urządzeń takich jak przełączniki, sterowniki, czujniki oraz pozwala na współpracę i wymianę informacji między tymi urządzeniami poprzez użycie standardowego protokołu komunikacyjnego. Sieć LON może z powodzeniem pracować na różnych mediach fizycznych (również mieszanych), takich jak:

- skrętka (kilka rodzajów komunikacji, możliwa realizacja dowolnej topologii połączeń),
- okablowanie energetyczne (we wnętrzu budynków, na zewnątrz budynków i dowolne inne),
- kanały na częstotliwościach radiowych.

W sieciach LON możliwe są szybkości transmisji informacji od 4,8 kb/s do 1,25 Mb/s, a jedna sieć może zawierać ponad 32 tysiące współpracujących elementów, czyli tzw. węzłów sieci. Dostęp do medium fizycznego odbywa się na zasadzie CSMA (*Carrier Sense Multiple Access*), tej samej jak w sieciach komputerowych NetWare, UNIX, w odróżnieniu od realizowanych dotąd sieci sterujących typu Master-Slave, Multi Master, Time Division Multiplexing, Token Ring, Token Bus.

Sercem każdego węzła sieci LON jest specjalnie do tego celu zaprojektowany układ mikroprocesorowy o nazwie NEURON®. Posiada on wbudowaną inteligencję w postaci procesorów sieciowych i aplikacyjnych, pamięci, systemu operacyjnego, protokołu komunikacyjnego, dedykowanych fizycznych portów wejścia/wyjścia, a wszystko to w jednym układzie scalonym o wymiarach 14x14x2.5 mm produkowanym na licencji firmy ECHELON przez dwie firmy - MOTOROLA i TOSHIBA.

Sieć LON posiada własny, stworzony specjalnie dla systemów sterowania protokół komunikacyjny nazwany LONTALK™. Protokół ten spełnia zalecenia modelu odniesienia dla połączeń systemów otwartych (*Open Systems Interconnection - OSI*) opracowanego przez Międzynarodową Organizację Standaryzacyjną (*International Standard Organization - ISO*). W ramach ISO protokół LONTALK dostarcza usług we wszystkich siedmiu warstwach modelu odniesienia OSI:

- zarządzanie kanałem fizycznym (warstwy 1 i 2),
- identyfikacja, adresowanie i wyznaczanie tras (warstwy 3 i 6),
- niezawodna łączność i efektywne wykorzystanie sieci (warstwy 2 i 4),
- system priorytetów transmisji (warstwa 2),
- działania zdalne (warstwa 5),
- potwierdzanie tożsamości (warstwy 4 i 5),
- zarządzanie siecią (warstwa 5),
- przesyłanie obcych ramek i interpretacja danych (warstwa 6),
- zgodność aplikacji (warstwa 7).

Jak łatwo zauważyć, sposób opracowania, instalacji, uruchamiania oraz testowania, konfiguracji i rekonfiguracji aplikacji systemu obiektowego o inteligencji rozproszonej znacznie się różni od sposobu podejścia w scentralizowanym odpowiedniku



takiego systemu chociażby dlatego, że mamy do czynienia z całą siecią sterującą, a nie pojedynczym komputerem centralnym.

W celu umożliwienia opracowania i uruchomienia aplikacji w węzłach sieci LON oraz całego, kompleksowego systemu sterowania LONWORKS firma ECHELON opracowała system uruchomieniowy pod nazwą LONBUILDER™. Posiadanie tego systemu jest niezbędne do tworzenia i uruchomienia sieci LON.

System LONBUILDER współpracuje z komputerem PC jako stacją centralną. Za jego pomocą można sprzętowo zasymulować jednocześnie do 24 węzłów sieci w celu opracowania i uruchomienia aplikacji sieciowej. System LONBUILDER ma budowę modułową, a jego moduły sprzętowe można dowolnie konfigurować. Każdy z modułów zawiera układ NEURON 3150 oraz 64 kB pamięci RAM dostępnej dla kodu wykonawczego, którego opracowanie w języku wysokiego poziomu NEURON C ułatwia oprogramowanie narzędziowe niezależnie od medium komunikacyjnego budowanej i testowanej sieci LON.

Firma ECHELON nadaje niektórym swoim partnerom status Niezależnego Twórcy Aplikacji LONWORKS - LID (*LONWORKS Independent Developer*). Są to firmy posiadające zestaw uruchomieniowy LONBUILDER oraz stosowną wiedzę i doświadczenie w budowie systemów obiektowych na bazie rozwiązania LONWORKS. Posiadanie przez firmę statusu LID pozwala na samodzielne opracowywanie i serwis sieci LON oraz sprzedaż usług i szkoleń w zakresie LONWORKS, korzystając z bezpośredniej pomocy specjalistów firmy ECHELON.

Instytut Komputerowych Systemów Automatyki i Pomiarów we Wrocławiu jest pierwszą i jak dotąd jedyną firmą w Polsce, która posiada własny system uruchomieniowy sieci LON LONBUILDER, a także jest w trakcie dopełniania ostatnich formalności z firmą ECHELON, twórcą rozwiązania LONWORKS dla sieci kontrolno-sterujących, zmierzających do otrzymania oficjalnego statusu LID. IKSAiP współpracuje z firmą ECHELON od maja 1993 roku. Na przełomie 1993-94 do współpracy dołączyły Instytut Automatyki Systemów Energetycznych z Wrocławia oraz Instytut Systemów Sterowania z Katowic.

Aktualne kierunki współpracy zmierzają do stworzenia silnej bazy usługowo - doradczej w zakresie LONWORKS w Polsce oraz popularyzacji zastosowań lokalnych sieci sterowania w zakładach i przedsiębiorstwach krajowych, do realizacji szerokiej gamy polskich wyrobów na bazie sieci LON.

## Literatura:

- [1] LONWORKS Engineering Bulletin - ECHELON 1992.
- [2] LONWORKS Interoperability Guidelines - ECHELON 1992.





BOŻENA PRZYBOROWSKA

INSTYTUT MASZYN MATEMATYCZNYCH WARSZAWA

## Transputery i OCCAM

### The Transputers and OCCAM

#### Streszczenie

Niniejsze opracowanie zawiera krótką specyfikację architektur równoległych ze szczególnym przedstawieniem modułów transputerów, które połączone w sieci tworzą formę systemu wieloprocessorowego z pamięcią rozproszoną. Omówiono bardziej szczegółowo język OCCAM, który opracowany specjalnie do celów przetwarzania równoległego, umożliwia tworzenie i synchronizację procesów równoległych, komunikację między nimi, a także sterowanie czasem rzeczywistym. Na zakończenie przedstawiono trendy rozwojowe układów transputerowych i oprogramowania.

#### Abstract

This article contents short specification of parallel computers' architecture with emphases placed upon transputer modules. These processors could be connected into nets and create the multiprocessing systems with the distributed memory. Then there were presented some details of the OCCAM language with stressing its particular ability to organize the synchronization and the communication among the parallel processes and the real time control. At last there were assumed trends of the development of transputer modules and their programming.

#### 1. Wstęp

Konwencjonalny procesor przetwarzający dane numeryczne działa sekwencyjnie. W ostatnich latach dokonuje się w świecie dynamiczny rozwój komputerów o architekturach odbiegających od klasycznego, sekwencyjnego modelu. Ponieważ naturalną cechą wielu zachodzących w naturze procesów jest równoległość, pojawiły się koncepcje wykorzystania wielu procesorów działających równoległe do przetwarzania algorytmów i wykonywania obliczeń na wielu strumieniach danych równocześnie.

Spśród liczących prób organizacji obliczeń w komputerach o architekturach równoległych szersze zastosowanie znalazły dwa typy architektur:

- SIMD (*Single Instruction Stream/Multiple Data Stream*) - przetwarzanie typu potokowego charakteryzujące się podziałem jednostki przetwarzającej na niezależne, kolejno po sobie następujące fragmenty, przez co może ona w tym samym czasie wykonywać kilka rozkazów realizujących tę samą operację na

zasadzie fabrycznej linii montażowej. Charakterystyczna jest tu synchroniczność pracy równoległej, gdyż na wszystkich strumieniach danych wykonywana jest ta sama operacja w tym samym momencie. Innym rozwiązaniem może być również podział na grupy rozkazów, które poprzez zwiokrotnione jednostki przetwarzające wykonywane są równoległe w ramach grupy;

- MIMD (*Multiple Instruction Stream/Multiple Data Stream*) - architektura wieloprocessorowa, gdzie praca równoległa realizowana jest w sposób asynchroniczny, a więc rozkazy wykonywane w danym momencie przez poszczególne procesory nie muszą być identyczne. Tego typu systemy występują ogólnie jako systemy ze wspólną pamięcią (*shared memory*), gdzie komunikacja odbywa się poprzez dostęp do wspólnych, dla komunikujących się procesorów, obszarów pamięci lub z pamięcią rozproszoną, gdzie każdy procesor ma własną pamięć i komunikacja między procesorami odbywa się na zasadzie przesyłania komunikatów (jak w sieciach komputerowych).

Wśród architektur równoległych zaczynają dominować systemy z przesyłaniem komunikatów, ze względu na możliwość zastosowania jako elementów przetwarzających stosunkowo tanich mikroprocesorów. Mimo że każdy z procesorów może posiadać niewielką moc obliczeniową, przyspieszenie obliczeń uzyskuje się poprzez zastosowanie dużej liczby takich procesorów.

Przedstawione architektury równoległe wykorzystywane do przeprowadzania skomplikowanych obliczeń naukowo-inżynierskich pojawiają się głównie jako komputery masywnie zrównoleglone o dużej mocy obliczeniowej. Ale architektury równoległe wkraczą również do systemów mniejszych, czego przykładem mogą być układy (karty) transputerów, będących formą systemu wieloprocessorowego z pamięcią rozproszoną.

## 2. Architektura i zastosowanie transputerów

### 2.1. Architektura

Transputer jest 16-bitowym lub 32-bitowym procesorem RISC opartym na technologii VLSI (*Very Large Scale Integration*). Standardowy moduł zawiera procesor, pamięć i linie komunikacji, a więc jest przygotowany układowo i programowo do tworzenia sieci wieloprocessorowych i obliczeń równoległych. Sieci transputerowe pracują w trybie procesów sekwencyjnych, komunikujących się poprzez synchroniczne, niebuforowane kanały transmisji. Zawarcie w jednym zintegrowanym układzie zarówno procesora, jak i pamięci powoduje, że przetwarzanie wewnątrz transputera jest wielokrotnie szybsze niż w standardowym procesorze. Natomiast konieczność podłączenia kart transputerowych do standardowego komputera ogranicza szybkość działania, umożliwiając dostęp tylko poprzez interfejs DMA (*Direct Memory Access*). Dlatego zastosowanie współbieżnych systemów transputerowych jest tym efektywniejsze, im wymaga więcej obliczeń, a mniej przesyłania danych do komputera głównego.



## 2.2. Procesory

Processor transputera wykorzystuje dostępność szybkiej pamięci na zintegrowanym układzie scalonym poprzez zastosowanie małej liczby rejestrów. Przy wykonywaniu procesów sekwencyjnych używanych jest sześć następujących rejestrów:

- wskaźnik obszaru roboczego, wskazujący miejsce przechowywania danych lokalnych,
- licznik rozkazów, wskazujący kolejną wykonywaną instrukcję,
- rejestr do formowania operandu instrukcji,
- rejestry arytmetyczne A, B i C tworzące stos i są rejestrami źródłowymi i docelowymi większości operacji arytmetycznych i logicznych. Przed załadowaniem rejestru A dotychczasowa zawartość rejestru B jest przesyłana do C, a A do B. Pobranie zawartości rejestru A powoduje przeladowanie zawartości B do A i C do B. Wyrażenia są obliczane przy operacjach na stosie i instrukcje działają bezpośrednio na stosie, nie zajmując się położeniem swoich operandów.

Pierwszym urządzeniem rodziny transputerów był transputer 32-bitowy, stalo-przecinkowy IMS T414 powstały w 1985 roku. Potem, poprzez kolejne modyfikacje, powstał moduł T800 kompatybilny w sensie łączówki, ale zawierający dodatkowo wbudowaną 64-bitową jednostkę operacji zmiennoprzecinkowych, instrukcje grafiki i dwukrotnie więcej pamięci niż IMS T414. Powstała szeroka paleta modułów firmy INMOS, zwanych TRAM (*TR*ansputer *M*odules).

## 2.3. Zastosowanie transputerów do przetwarzania w czasie rzeczywistym

Transputery dostarczają narzędzi sprzętowych i w formie mikrokodów dla obsługi zagadnień przetwarzania równoległego i komunikacji. W układach tych zawarte są również połączenia do komunikacji między transputerami i dwa zegary sprzętowe, które mogą być użyte do mierzenia przedziałów czasowych lub przetwarzania w czasie rzeczywistym.

Językiem przetwarzania równoległego integralnie związanym z transputerem jest OCCAM, który umożliwia tworzenie i synchronizację procesów równoległych, a także sterowanie czasem rzeczywistym.

## 2.4. Zastosowanie transputerów w grafice

Jedną z najszybciej rozwijających się dziedzin systemów komputerowego przetwarzania jest grafika interakcyjna i komputerowe tworzenie obrazu, symulacja obiektów, obserwacja ich zmian. Wymaga to odpowiedniej szybkości i przetwarzania wielkiej ilości danych, jako że obiekt musi być przedstawiony realistycznie, we właściwej perspektywie, wymiarze i cieniowaniu. Konkretnie problem przedstawiania i modelowania obiektów graficznych były rozwiązywane na transputerach, np. T800 - 32-bitowym procesorze, z pamięcią 4 kB RAM i zmiennym przecinkiem, na jednym układzie scalonym, połączonym interfejsem zewnętrznym z pamięcią 4 GB.

### 3. Sieci transputerowe

Transputer, będący mikroprocesorem o wysokiej mocy przetwarzania, posiada cztery szeregowe łącza komunikacyjne przesyłające informacje z szybkością od 10 do 20 Mb/s. Łącza te pozwalają na tworzenie dowolnych sieci transputerowych o różnorodnej konfiguracji. Ponieważ w języku OCCAM każdy proces jest procesem niezależnie wykonywalnym, więc zestaw procesów może być wykonywany zarówno przez pojedynczy procesor, jak i przez kilka procesorów połączonych ze sobą w różnych kombinacjach. Programista ma za zadanie odpowiednio rozdzielić procesy pomiędzy procesory w sieci transputerów. Topologia sieci powinna zostać dopasowana do liczby transputerów i konkretnej aplikacji. Tworzenie zestawu procesów, z których każdy działa w sieci na indywidualnym transputerze i jest zaopatrzony w kod bootstrap'u umożliwiającą uruchamianie z procesora głównego i ładowanie do sieci transputerów, nazywany konfigurowaniem.

### 4. Oprogramowanie

Cechy urządzeń transputerowych z wieloma interakcyjnie działającymi komponentami mogą być wyrażone tylko przy pomocy języka, który może opisać charakterystyczne cechy równoległości i komunikację. Język programowania OCCAM umożliwia tworzenie programów aplikacyjnych jako zbioru procesów, które wykonują się równolegle i komunikują poprzez kanały.

W takim opisie każdy proces opisuje jeden element zagadnienia, a każdy kanał - połączenia między elementami. Programista dostaje narzędzie umożliwiające synchronizację i komunikację między procesami, a inżynier oprogramowania i projektant systemowy może do projektowania systemów sprzętowych i programowych wykorzystywać programy równolegle napisane w języku OCCAM, a następnie implementować je przy użyciu transputerów.

#### 4.1. Koncepcja programowania równoległego

OCCAM upraszcza pisanie programów równoległych przejmując na siebie czynności synchronizowania poszczególnych procesów i komunikacji między nimi. Za przykład niech posłuży wykonanie swetra, co można w języku OCCAM przedstawić następująco:

```
SEQ                                -- to jest komentarz
  PAR
    ...wykonaj przód
    ...wykonaj tył
    ...wykonaj prawy rękaw
    ...wykonaj lewy rękaw
  ...uszyj sweter
```

Oznacza to, że części swetra będą wykonywane równolegle, ale szycie nastąpi sekwencyjnie, dopiero po wykonaniu wszystkich części.

Komunikacja i synchronizacja pomiędzy częściami programu jest wbudowana w strukturę języka, a wysyłana jest jedynie informacja o gotowości odbiorcy i nadawcy.



Jeśli jedna z części programu zakończy pracę wcześniej, będzie oczekiwać na zakończenie pracy przez inne części, nie wymagając żadnych specjalnych instrukcji od programisty. Musi on tylko zadbać o uniknięcie sytuacji impasowej przez upewnienie się, że każda z części programu jest wykonywalna i ma szansę kiedyś się zakończyć.

## 4.2. Procesy i kanały

Części programu w języku OCCAM są nazywane procesami. Proces rozpoczyna się, wykonuje pewną ilość akcji przetwarzania i kończy się. Ta definicja pasuje też do programu sekwencyjnego, ale w OCCAM więcej niż jeden proces może być wykonywany równocześnie i procesy mogą przysyłać między sobą informacje.

Zapewnienie komunikacji między procesorami jest cechą programowania w języku OCCAM. Używa on zmiennych do przechowywania wartości, natomiast kanałów do komunikacji i przekazywania wartości. Kanał zachowuje się jak rura łącząca dwa procesy. Pojedynczy kanał łączy tylko dwa procesy i działa tylko w jednym kierunku. W związku z tym w celu komunikacji dwustronnej we/wy konieczne są dwa kanały. Przesyłanie polega na kopiowaniu wartości zmiennej poprzez kanał jako wartości wejściowej lub wyjściowej. OCCAM używa symboli "!" do oznaczenia wyjścia oraz "?" do oznaczenia wejścia:

A ! 2	⇒	A ? B
(proces 1)		(proces 2)

gdzie A jest kanałem, a B zmienną i oznacza "wysłanie 2 do kanału A" i "pobranie wartości z kanału A do zmiennej B".

Do chwili gdy oba procesy są niezależne, mogą być one wykonywane w różnym czasie. Przesyłanie wartości z jednego kanału do drugiego jest możliwe dopiero w chwili, gdy oba procesy są gotowe. Jeśli wyjście w procesie 1 wykona się przed wykonaniem operacji wejścia w procesie 2, proces 1 automatycznie oczekuje gotowości procesu 2 do przesłania wartości, i vice versa, jeśli wejście w procesie 2 będzie gotowe przed wysłaniem wartości przez proces 1, proces 2 automatycznie oczekuje na pojawienie się wartości na wyjściu procesu 1. Nie ma możliwości "zgubienia" wartości przesyłanej.

Są również dwie następne cechy charakterystyczne różniące kanał od zmiennych:

- kanał może przysyłać wartości zarówno pomiędzy dwoma procesami wykonującymi się na tym samym komputerze, jak i między dwoma procesami na różnych komputerach. W pierwszym przypadku kanał będzie raczej miejscem w pamięci, czyniąc w rodzaju zmiennej. W drugim przypadku kanał będzie reprezentować połączenie sprzętowe, np. poprzez transputery lub inne łącze szeregowe. Oba przypadki są w języku OCCAM reprezentowane identycznie. Kanały w języku OCCAM definiują komunikację abstrakcyjną, niezależną od implementacji. Program może być opracowywany na pojedynczy procesor, a po uruchomieniu można zadecydować o rozdysponowaniu procesów pomiędzy różne komputery poprzez dodanie paru prostych deklaracji na początku programu;
- jeśli proces wejściowy nie otrzymuje spodziewanej danej, oczekuje na nią. Również wyjście nie wysyła wartości do chwili, gdy wejście zgłosi gotowość.

Wprowadza to do programowania czynnik czasu, jednakże niezależny od programisty.

Implementacja języka OCCAM wymaga zadeklarowania zmiennych i kanałów przed ich użyciem.

## 5. Podstawy języka OCCAM

### 5.1. Składnia i format programów

Każde zdanie programu w języku OCCAM zajmuje pojedynczą linią, a istotną część składni języka stanowi forma wcięcia każdego zdania. Każdy przedstawiony proces występuje w osobnej linii i przesunięty jest o dwie spacje w stosunku do poprzedzających je słów kluczowych. Najważniejsze słowa kluczowe to:

SEQ	- proces sekwencyjny
PAR	- proces równoległy
ALT	- alternatywa
CHAN OF	- typ kanału
STOP	- proces stop
SKIP	- proces skip
TIMER	- typ zegara
CASE	- selekcja
FOR	- licznik
IF	- warunek
FROM	- baza

Komentarz w języku OCCAM poprzedzany jest przez podwójny symbol myślnika "--", rozciąga się do końca linii i nie może być wcięty mniej niż następujące po nim zdanie (każde następne wcięcie wynosi zawsze dwie spacje).

Długie zdanie może być przeniesione do następnej linii bezpośrednio po:

- operatorach np. "+", "-", "\*", "/" itd.
- przecinku ","
- średniku ";"
- znaku przypisania "!="
- słowach kluczowych "IS", "FROM" lub "FOR".

Nazwy używane w programach muszą rozpoczynać się od litery i mogą składać się z ciągu znaków alfanumerycznych i kropek. Nazwy nie mają ograniczenia długości, ale są rozróżniane małe i duże litery, np. nazwa "Terminal" i "terminal" to dwie różne nazwy. Jednakże słowa kluczowe w języku OCCAM są zawsze pisane dużymi literami. Ich nazwy są zarezerwowane i nie mogą być używane przez programistę.

### 5.2. Procesy proste

Wszystkie programy w języku OCCAM budowane są przy użyciu trzech rodzajów prostych procesów: przypisania, wejścia i wyjścia.



### Proces przypisania

Przypisanie zmienia wartość zmiennej. Symbolem jest ":=". Symbol "=" oznacza test na równość, a nie przypisanie., np.

`x := y+2`

przyporządkowuje zmiennej `x` wartość `y+2`.

W języku OCCAM może występować przypisanie wielokrotne, tzn. jednoczesne nadanie wartości wielu zmiennym, np. `a, b, c:= 2, x+3, y+2`. Wyrażenia występujące po prawej stronie znaku przypisania są obliczane, a ich wartości są odpowiednio przypisywane zmiennym `a, b, c`.

### Proces wejścia

Proces wejścia pobiera wartość z kanału i przypisuje ją zmiennej (nie stałej lub wyrażeniu).

`chan3 ? ala`

pobiera wartość z `chan3` i wprowadza do zmiennej `ala`.

Wartość wejścia i zmienna, do której ma być ono przypisane, muszą być tego samego typu. Proces wejścia oczekuje i nie wykonuje się do chwili, aż będzie gotowy odpowiadający proces wyjścia dla tego samego kanału.

### Proces wyjścia

Proces wyjścia przesyła wartość, określoną po prawej stronie wyrażenia, do kanału.

`chan3 ! 2`

wysyła wartość `2` do kanału `chan3`.

Wartość wysyłana do kanału może być każdym elementem, jaki można przypisać zmiennej, a więc zmienną lub wyrażeniem, czy też wyrażeniem zawierającym zmienne.

Proces wyjścia nie wykona się aż do chwili, gdy będzie gotowy proces wejścia w tym samym kanale.

### Komunikacja

Komunikacja poprzez kanał może odbywać się, gdy gotowe są oba procesy wejście i wyjście. Jeśli w czasie wykonywania programu proces wejścia zakończy się zanim odpowiadający mu proces wyjścia będzie dostępny, wejście oczekuje na gotowość wyjścia i odwrotnie. Wartość przesyłana kanałem zastępuje zmienną wejściową, a wartość zmiennej wyjściowej pozostaje niezmieniona.

Komunikacja poprzez kanał jest zsynchronizowana.

Programy w języku OCCAM budowane są z prymitywnych procesów.

### SKIP i STOP

Proces **SKIP** rozpoczyna się, nie wykonuje jakiegokolwiek akcji i kończy się. Proces ten może być używany w przypadkach, gdy nie ma się wykonać nic lub w miejscu obsługi programu, który będzie wstawiony później.

Proces **STOP** rozpoczyna się, ale nie wykonuje się i nigdy nie kończy, uniemożliwiając zakończenie sekwencji. Ma on własności procesu błędnego i może np. zastępować proces obsługi błędów.

### Zakończenie i zatrzymanie

Proces rozpoczyna się, wykonuje i kończy. Zakończenie procesu następuje po wykonaniu wszystkich czynności. Proces, który nie może z jakiegoś powodu wykonywać się, jest zatrzymywany przez **STOP**, co nie oznacza zakończenia procesu, gdyż taki proces nigdy się nie skończy (może np. czekać na zdarzenie, które nigdy nie nastąpi).

## 5.3. Procesy złożone

Procesy proste mogą być łączone w większe struktury, przez ich składanie i określenie budowanej konstrukcji: sekwencyjnej, równoległej, alternatywnej, warunkowej lub wyboru.

### Sekwencja

Konstrukcja **SEQ**, stwarza układ jak w każdym konwencjonalnym programie, składa procesy w konstrukcję, w której jeden proces rozpoczyna się po zakończeniu poprzedniego. Ponieważ jednak **OCCAM** przewiduje różne tryby pracy, przetwarzanie sekwencyjne musi być obowiązkowo zaznaczone słowem **SEQ**. Proces następujący po słowie kluczowym musi być wcięty o dwie spacje. Jest to ogólna zasada zaznaczania nowego procesu w konstrukcji programu.

### Równoległość

Równoległość łączy procesy przetwarzane równocześnie. Konstrukcja **PAR** oznacza wykonywanie czynności równoległe, tak że wszystkie procesy tej konstrukcji rozpoczynają się jednocześnie, a cały proces **PAR** zakończy się, kiedy wszystkie jego elementy zakończą pracę. Wartości pomiędzy procesami równoczesnymi przesyłane są przez kanały komunikacyjne przy użyciu procesów wejścia i wyjścia.

### Warunek

Wybór warunkowy w języku **OCCAM** umożliwia konstrukcja **IF**. Konstrukcja ta może zawierać dowolną liczbę procesów, każdy poprzedzony testem będącym wyrażeniem boolowskim. Każdy taki proces składowy (test plus wyrażenie) nazywamy "wybozem". Każde wyrażenie boolowskie jest obliczane szeregowo, a zostanie wykonany pierwszy proces, dla którego wyrażenie to będzie prawdziwe (test da wartość **TRUE**). Jeżeli żaden warunek konstrukcji nie jest spełniony, zachowuje się ona jak proces **STOP**. Można tego uniknąć dodając na końcu listy wyborów konstrukcji **IF** warunek **TRUE**, który będzie się wykonywać, jeśli wszystkie poprzednie nie będą spełnione (jeśli chcemy, by nie się wtedy nie działo, po warunku **TRUE** wprowadzamy proces **SKIP**). Konstrukcje **IF** mogą być zagnieżdżone.

### Selekcja

Konstrukcja selekcji - **CASE** umożliwia dokonywanie wyboru w zależności od dopasowania wartości zmiennych do klucza selektora. Konstrukcja składa się z dowolnej liczby procesów, każdy poprzedzony listą wyrażień, które w wyniku działania in-



strukcji stają się jednym procesem. Wykona się tylko jeden proces - pierwszy, który zawiera wyrażenie o tej samej wartości, co zmienna wyboru. Aby uniknąć przypadku, kiedy żadna z opcji (wyrażenie plus proces) nie spełni warunku i program zachowa się jak proces STOP, można dodać opcję ELSE wykonującą się, gdy żadna z opcji konstrukcji CASE nie jest równa kluczowi selektora.

### Pętla

OCCAM umożliwia tworzenie pętli poprzez powtarzanie procesu określoną ilość razy lub do chwili spełnienia warunku. Pętla warunkowa tworzona jest za pomocą WHILE. Jako stałe wartości logiczne używane są w języku OCCAM TRUE i FALSE. Efektem dodatkowym pętli WHILE jest przechowywanie wartości pomiędzy wejściem a wyjściem.

### Procesy alternatywne

Proces alternatywny to rodzaj konstrukcji wyboru nie stosowany w innych językach programowania, pozwalający dokonać wyboru w zależności od stanu kanałów. Proces ALT łączy w swojej konstrukcji dowolną liczbę składników. Najprostsza konstrukcja zawiera procesy do wykonania uzależnione od procesów wejścia. Wybór jest tu uzależniony od czasu, gdyż program oczekuje na pierwszy kanał, w którym pojawi się wartość wejściowa i wykona proces związany w konstrukcji ALT z tym kanałem. Każda z alternatyw może dodatkowo - oprócz oczekiwania na wejście - zawierać test, który musi przybrać wartość TRUE, aby proces związany z danym kanałem być wykonany. Łączenie warunków przedstawiane jest za pomocą znaku "&". Jest to konstrukcja bardzo silna, pozwalająca na łączenie i przełączanie kompleksowej sieci kanałów. Konstrukcje ALT mogą być zagnieżdżone.

## 5.4. Elementy języka OCCAM

### Typy danych, znaki, deklaracje i zasięg

Każdy obiekt użyty w programie musi być najpierw zdefiniowany. Programy działają na zmiennych, kanałach i zegarach. Zmienne przyjmują wartości nadane przez przypisanie lub wejście. Kanały przesyłają wartości danych. Zegary generują wartości reprezentujące czas.

Wartości są klasyfikowane według typów określających zbiór wartości, jakie może przyjmować dany obiekt. Przykładowe typy wartości:

INT, INT16, INT32, INT64	- liczby całkowite ze znakiem z różnych przedziałów
REAL, REAL32, REAL64	- liczby rzeczywiste zmiennoprzecinkowe
BYTE	- liczby całkowite od 0 do 255
BOOL	- wartości boolowskie

Wiele zmiennych jednego typu może być określanych jednocześnie w formie listy o elementach oddzielonych przecinkami (np. INT ala,jim,fred).

Obiekty mające w języku OCCAM wartości przyjmują jedną z form:

- zmienne - nazwy symboliczne
- stałe - nazwy symboliczne posiadające stałe wartości
- literały - tekstowe reprezentacje wartości
- indeksy - wartości indeksu powtórzenia

Nazwy obiektów są lokalne w procesie, który następuje po ich deklaracji, a nie odnoszą się do innych procesów. Dwukropek, który kończy deklarację typu, w efekcie ogranicza jej zasięg do procesu, który po niej następuje. Deklaracja występuje na tym samym poziomie co proces.

Wartości zmiennych są niezdefiniowane do chwili przypisania lub pobrania wartości z wejścia. Po zakończeniu przetwarzania wartość zmiennej uzyskana w procesie traci sens na zewnątrz procesu. W celu przechowania wartości zmiennej przy wykonywaniu różnych procesów, trzeba ją zadeklarować jako zmienną o zasięgu zewnętrznym.

OCCAM posiada notację opisu danych tekstowych. Poszczególne znaki zapisywane są w pojedynczym cudzysłowie 'a'. Ciąg tekstowy zapisywany jest w podwójnym cudzysłowie "ala" (równoważne macierzy o elementach ['a','l','a']).

Znaki specjalne są w notacji poprzedzane \*, np. \*n oznacza nową linię, a \*t tabulację. Ogólnie każdy znak może być reprezentowany przez znak \* poprzedzający jego kod heksadecymalny.

Znaki ciągu mogą być przypisane macierzy i wybierane przy pomocy indeksów, jak również części ciągów mogą być wybierane jako segmenty macierzy.

Ciągi znaków mogą być przesyłane na wyjście jako macierze. Jest to przysyłanie w formie bloku odpowiednio dużego, aby pomieścić przesyłany ciąg i odpowiednio ograniczanego przy przysyłaniu np. poprzez SIZE (*message*), oczywiście przy zastosowaniu właściwego protokołu przysyłania.

## Stałe

Nadawanie wartości stałym odbywa się poprzez deklarację:

**VAL** nazwa typu **IS** wartość:

## Zmienne

Deklaracja zmiennej nadaje jej nazwę i deklaruje typ. Jest zakończona ":", np.:

**INT** n:

**BYTE** char:

OCCAM dopuszcza deklarowanie w pojedynczej linii wielu zmiennych tego samego typu, np.:

**REAL64** a,b,c:

Nazwy zmiennych w deklaracjach wielokrotnych są oddzielane przecinkami.

## Literal

Literal jest tekstową reprezentacją wartości i posiada typ. Liczba, np. 42, jest liczbą typu **INT** i przedstawia wartość dziesiętną lub heksadecymalną, gdy poprzedzona jest znakiem hash (#). Znak ujęty w apostrofy ('z') ma wartość typu **BYTE**, a ciąg jest macierzą bajtów, a więc np. "zen" jest macierzą typu [3]**BYTE**. Literalowe wartości innych typów są ustawiane przez jawne ustawienie typu danych:

- 42(**BYTE**)                      liczba bajtowa
- 'T'(**INT**)                      liczba całkowita
- 16.55(**REAL64**)              64-bitowa liczba zmiennoprzecinkowa



Literaly TRUE i FALSE reprezentują wartości boolowskie: prawdę i fałsz.

### Kanały

W kanałach jest prowadzona niebuforowana, jednokierunkowa transmisja danych między dwoma współbieżnymi procesami. Format i typ danych przesyłanych przez kanał określa protokół kanału. Kanały posiadają typ protokołu **CHAN OF**, który określa typ danych i strukturę wartości, które będą przesyłane tym kanałem.

Kanały deklarowane są w taki sam sposób jak zmienne, np.:

**CHAN OF BYTE screen:**

deklaruje kanał o nazwie *screen* z protokołem typu **BYTE**. Wiele kanałów z tym samym protokołem można deklarować razem. Protokoły kanałów zostaną opisane w dalszej części.

### Zegary

Zegary generują wartości reprezentujące czas i określane są typem **TIMER**.

### Skróty

Skrót jest wyrażeniem definiującym używanym w celu nadania nazwy dowolnemu kompleksowemu wyrażeniu arytmetycznemu w języku OCCAM., np.:

**VAL exp IS ((x+y) / (z\*128)) :**

definiuje "exp" jako równoważne wyrażeniu po prawej stronie. Dwukropek kończący specyfikacją oznacza, że skrót ma znaczenie lokalne w zasięgu danego procesu. Wszystkie zmienne użyte w wyrażeniu muszą mieć wartość stałą w zasięgu skrótu, tak że skrót zachowuje się również jak stała. Przed nazwą skrótu może zawsze pojawić się definicja typu, która jednak w razie jej braku jest automatycznie nadawana przez translator.

### Macierze

Macierz jest zbiorem obiektów podobnego typu, objętych wspólną nazwą. Każdy obiekt, określony jako element macierzy, jest dostępny na podstawie numeru jego pozycji. Elementem macierzy może być nazwa, nazwa indeksowana lub segment innej macierzy. Zmienne tworzące macierz są deklarowane w ten sam sposób jak zmienne każdego typu, ale z wymienioną liczbą elementów umieszczoną w nawiasach kwadratowych poprzedzających specyfikację. Wielkość macierzy musi być określona podczas kompilacji. Element macierzy używany w programie wybierany jest przez nazwę macierzy i umieszczony za nią w nawiasach kwadratowych indeks elementu.

Typ macierzy odzwierciedla zarówno jej wielkość jak i typ elementów. Element będący nazwą, identyfikuje zmienną lub macierz zmiennych, kanał lub macierz kanałów, zegar lub macierz zegarów. Jeśli macierz będzie elementem wejścia lub wyjścia, odpowiedni kanał musi być zadeklarowany jako przystosowany do przesyłania danych odpowiedniego typu macierzowego:

**CHAN OF [20] INT nazwa macierzy :**

Możemy określać segmenty macierzy w formie:

[macierz **FROM** numer elementu **FOR** licznik]

Taki segment składa się z pewnej liczby kolejnych elementów macierzy. Segmenty macierzy mogą być używane jako wejście lub wyjście. Mogą również być przypisywane jako ekwiwalent całej macierzy, ale tylko gdy nowa macierz jest tego samego typu co segment. Tablice w języku OCCAM są macierzami generującymi wartości.

Jeśli często używamy w programie określonego segmentu macierzy, można zdefiniować go jako skrót. Można zmieniać wartość obiektu, do którego odwołujemy się poprzez skrót, przez wejście lub przypisanie. Jednak skrót definiujący element macierzy zachowuje się w zasięgu skrótu jak element macierzy o stałym indeksie. Skrót może być też użyty do określenia macierzy stałych elementów i wtedy nazwa musi być poprzedzona słowem kluczowym **VAL**. Elementy są dostępne, tak jak elementy macierzy, ale oznaczają dostęp do stałych wartości elementów:

**VAL** nazwa macierzy **IS** [1,2,4,8,16,32] :

Macierze dwuwymiarowe oznaczane są indeksami określającymi pozycję elementu jako wiersz i kolumną oraz deklarowane są poprzez podanie wymiarów w kolejnych nawiasach kwadratowych przed nazwą macierzy.

## Procedury

Procedura jest procesem posiadającym nazwę, która pozwala wykorzystywać tę procedurę w innych procesach. W definicji procedury po słowie kluczowym **PROC** i nazwie występuje proces będący ciałem procedury. Pojawienie się nazwy procedury w programie jest równoznaczne z "żądaniem" jej wywołania i powoduje wykonanie ciała procedury. Zakończenie definicji procedury jest oznaczone dwukropkiem, który musi pojawić się w tej samej kolumnie co litera "P" określenia **PROC**.

Procedura staje się narzędziem bardzo użytecznym poprzez wprowadzenie parametrów umożliwiających wywołanie procedury z różnymi ich wartościami. Nazwa pojawiająca się w definicji po nazwie procedury jako (**VAL** typ nazwa) jest parametrem formalnym procedury. Parametr formalny może być dowolnego typu, również **CHAN**, ale w przeciwieństwie do skrótu tutaj określenie typu parametru formalnego jest konieczne, choć zachowuje się on jak skrót przypisany do ciała procedury. Gdy ciało procedury jest zastąpione w procesie jej wywołaniem, nazwa parametru formalnego staje się skrótem wartości nazywanej parametrem aktualnym. W definicji procedury może wystąpić kilka parametrów formalnych oddzielonych przecinkami. Każdemu parametrowi formalnemu musi odpowiadać w wywołaniu parametr aktualny.

W języku OCCAM parametr jest przekazywany jako nazwa zmiennej i wszystkie operacje wykonywane w procedurze na parametrze formalnym dotyczą zmiennej, która w ten sposób może zmieniać swoją wartość (nie ma wywołania przez wartość jak w innych językach programowania). Jeśli nie chcemy, aby procedura wpływała na wartość parametru, musimy deklarować go jako zmienną lokalną i na niej manipulować lub przekazując parametr przez **VAL** używać go jako wartości stałej. Zmienna użyta w procedurze musi być zdefiniowana w programie, przed wywołaniem procedury.

## Funkcje

Funkcja jest procesem, który zwraca wartość. OCCAM określa ściśle zasady konstrukcji funkcji, która jest procesem zwracającym wartość, zwanym "value process".



Definicja:

typ **FUNCTION** nazwa ((,parametry formalne))

specyfikacja:

**VALOF**

proces

**RESULT** wyrażenie

Specyfikacja określa typ zmiennej będącej rezultatem funkcji. Parametry formalne mogą być wyłącznie typu **VAL**. Konstrukcje **PAR** i **ALT** jak również wejście i wyjście nie mogą być używane wewnątrz funkcji. Przypisania można dokonać tylko zmiennej zadeklarowanej wewnątrz funkcji. Inne zmienne (swobodne) mogą być odczytywane, ale nie przypisywane. W zasięgu funkcji mogą być używane jedynie procedury przestrzegające powyższych zasad. Funkcja może być użyta w każdym miejscu, gdzie może pojawić się wyrażenie arytmetyczne.

## 5.5. Arytmetyka w języku OCCAM

### Wyrażenia

Wyrażenia są konstruowane z operandów, operatorów i nawiasów. Najprostszymi wyrażeniami są literaly i zmienne. Operand jest elementem typu dane, literalem, tablicą lub innym wyrażeniem w nawiasach. Wyrażenie generuje wynik, a jego rezultat ma wartość i określony typ danych. W przypisaniu typ wartości wyrażenia musi być taki sam jak typ wartości zmiennej, do której przypisana jest wartość tego wyrażenia.

### Tablice

Tablica tworzy macierz wartości z pewnej ilości wyrażen, które muszą dostarczać wartości tego samego typu. Wartość każdej składowej macierzy jest wartością odpowiedniego wyrażenia. Tablica jest pewną ilością wyrażen tego samego typu oddzielonych przecinkami i zamkniętych w nawiasy kwadratowe.

### Operacje

Podstawowe operacje arytmetyczne mogą być wykonywane na danych typu **INT** lub **REAL**. Wszystkie operatory – "+", "-", "\*", "/" (dzielenie całkowite) oraz "REM" (reszta z dzielenia) – mają taki sam priorytet w języku OCCAM, należy więc w wyrażeniach arytmetycznych używać nawiasów.

Arytmetyka modulo  $2^{**}$ , gdzie liczba bitów jest typu **INT**, posługuje się operatorami:

<b>PLUS</b>	-- dodawanie
<b>MINUS</b>	-- odejmowanie
<b>TIMES</b>	-- mnożenie
<b>MOSTNEG</b>	-- największa ujemna
<b>MOSTPOS</b>	-- największa dodatnia

Mogą być też wykonywane standardowe operacje logiczne na poszczególnych bitach i wykorzystywane operatory relacji.

## 5.6. Komunikacja poprzez kanały

### Protokół prosty i sekwencyjny

Przy przetwarzaniu równoległym procesy przebiegają równolegle, ale niesynchronicznie, a synchronizację wymusza komunikacja między kanałami. Kanał musi określać jeden kierunek przesyłania, a więc w celu komunikacji dwukierunkowej między dwoma procesami należy zadeklarować dwa kanały.

Kanały muszą mieć określony typ, ponieważ mogą przysyłać dowolnego typu dane. Ważne jest, aby kanałami mogły być przysyłane dane różnych typów i można to osiągnąć poprzez grupowanie typów i tworzenie "protokołu" komunikacji. Protokół taki specyfikuje sekwencję typów obiektów, które mogą być przysyłane kanałem, a więc opisuje format komunikatów złożonych z grup typów danych języka OCCAM. Deklaracja kanału przysyłającego prosty typ danych wygląda następująco:

**CHAN OF** nazwa protokołu :

Najprostszy protokół może być podstawowym (prosty) typem danych. Jednak może być on bardziej kompleksowy, utworzony w formie sekwencji prostych protokołów i określający jakie sekwencje wartości i typy będą mogły być przysyłane kanałem. Wtedy występuje definicja protokołu:

**PROTOCOL** nazwa **IS** typ; typ; typ :

podająca listę przysyłanych typów danych oraz deklaracja kanału, którym dane odpowiadające temu protokołowi będą przysyłane:

**CHAN OF** nazwa protokołu nazwa kanału :

### Protokół przesyłania macierzy

Przy omawianiu macierzy podkreślono, że macierze mogą być przysyłane kanałami o protokołach tego samego typu, co stwarza ograniczenia, zważywszy, że macierze o dowolnych rozmiarach mogą być przysyłane jako parametry w procedurach. Dlatego istnieje protokół przesyłania macierzy pozwalający przesłać jej wymiar przed nazwą.

typ1 :: typ2

oznacza, że kanał może przysyłać pary wartości składających się z wartości typ1 określających wymiar, po którym następuje liczba elementów określonego typu macierzy typ2.

**CHAN OF INT** :: [INT] nazwa macierzy :

oznacza umożliwienie przesyłania przez kanał tablic o elementach całkowitych o dowolnym wymiarze.

**PROTOCOL** nazwa **IS** **BYTE::[]** **BYTE;** **BYTE::[]** **BYTE :**

pozwała przysyłać pary macierzy o elementach bajtowych, co może być wykorzystane do przesyłania ciągów reprezentujących tekst, a więc np. par słów.



### Protokół wariantowy

Protokół sekwencyjny specyfikuje format przesyłanej informacji, jednak w celu przesyłania informacji o różnych formatach należy użyć protokołu wariantowego. Jest to zbiór różnych protokołów pogrupowanych za pomocą instrukcji CASE.

```

PROTOCOL nazwa protokołu
CASE
  a; INT
  b; BYTE; INT
  c; BYTE; BYTE

```

określa protokół, gdzie a, b, c są oznacznikami, a deklaracja kanału

```

CHAN OF nazwa protokołu nazwa kanału :

```

oznacza, że kanał o zadanej nazwie może przysyłać informacje w formatach o różnych wariantach a, b lub c, ale należy jednak wiedzieć, który z wariantowych typów będzie przesłany. Dlatego OCCAM przewiduje możliwość użycia konstrukcji CASE w celu wykonania procesów odpowiadających otrzymaniu na wejściu informacji typu wariantu a, jak i wariantu b. W celu kontrolowania przebiegu procesu taka konstrukcja CASE może być użyta w konstrukcji PAR lub ALT.

Jeśli tylko jeden znany komponent protokołu wariantowego pojawia się na wejściu, można użyć konstrukcji wejścia ustalonego.

### 5.7. Replikatory

Jedną ze znaczących cech języka OCCAM jest umożliwienie konstruowania dodatkowo, oprócz macierzy danych i kanałów, macierzy procesów.

W każdej z konstrukcji SEQ, PAR, ALT oraz IF urządzenie nazywane replikatorem pozwala tworzyć macierz podobnych procesów odpowiadającego rodzaju. W konstrukcji replikacji indywidualne procesy mogą być dostępne poprzez indeks replikatora w ten sam sposób jak elementy tablicy. Oto forma replikatora :

```

REP indeks = baza FOR licznik
  proces

```

gdzie REP jest jednym z procesów SEQ, PAR, ALT lub IF. Jeśli licznik replikatora jest zerem, proces zachowuje się jak konstrukcja bez składników (SEQ jak SKIP, a IF oraz ALT jak STOP).

#### Replikator SEQ

Replikator zachowuje się jak pętla WHILE, ale nie wymaga specyfikacji zwiększania i szacowania zmiennej indeksowej w każdym przebiegu, indeks replikatora zwiększa się automatycznie od wielkości bazowej tyle razy, ile wynosi wielkość "licznik".

Może być użyty do wypełnienia macierzy:

```

SEQ i=0 FOR SIZE big.array
  input ? big.array[i]

```

co da efekt:

**SEQ**

```
input ? big.array[0]
input ? big.array[1]
.....
input ? big.array[(SIZE big.array)-1]
```

### Replikator PAR

Replikator **PAR** buduje strukturę procesów równoległych i pozwala na eksploatację wielu równoczesnych procesów przy użyciu techniki "nurociągu" (*pipeline*). Konstrukcja ta składa się z pewnej ilości procesorów, z których każdy jest zdolny do przetworzenia jednego kroku. Dane wprowadzane są na wejście pierwszego procesora, a końcowy rezultat odbierany na wyjściu. W międzyczasie na każdym pośrednim stopniu wykonuje się jeden krok przetwarzania przesyłanych danych. W ten sposób wiele procesów składających się z kolejnych stopni wykonuje się jednocześnie. Kolejne kroki nakładają się w czasie i wykonywane są w tempie dopływu nowych danych, bez oczekiwania na zakończenie przetwarzania wszystkich kroków procesu.

### Replikator ALT

Replikator **ALT** jest strukturą identycznych alternatyw, z których każda jest przełączana przez wejście z kanału. Proces zapisany za pomocą replikatora **ALT**, jako pobranie z  $n$  kanałów, wygląda następująco:

```
ALT i=0 FOR n
  kanwe[i] ? x
  kanwy ! x
```

gdzie  $n$ =wartość licznika. Zachowuje się tak jak multiplexer, pobierając i łącząc dane z każdego z  $n$  kanałów w miarę ich pojawiania się i wysyłając informację do jednego kanału wyjściowego. Pozwala to zbudować sieć przełączającą dowolnego kształtu i rozmiaru.

### Replikator IF

Ten replikator tworzy konstrukcję warunkową o zadanej ilości wyborów podobnej konstrukcji. Może być użyty np. do znalezienia pierwszego wystąpienia zadanego znaku w ciągu.

## 6. Przetwarzanie w czasie rzeczywistym

Konieczna jest możliwość pomiaru czasu w języku OCCAM, aby uzyskać równoległość wykonywania procesów i ich synchronizację.

### Zegary

Możemy deklarować obiekty typu **TIMER**



kilka procesów konstrukcji **PAR** może pobierać wejście z tego samego timer'a, co jest zabronione przy używaniu zwykłych kanałów.

### Opóźnienia

Opóźnienie deklarowane przez słowo kluczowe **AFTER**, po którym następuje wyrażenie reprezentujące czas, służy do opóźnienia odczytu procesu wejścia, aż do chwili osiągnięcia określonej wartości czasu. **AFTER** może również być użyte jako operator porównania generujący wartość **true**.

$x \text{ AFTER } y$  jest równoważne  $(x \text{ MINUS } y) > 0$

Operatory **MINUS** i **PLUS** używane są zamiast znaków "-" i "+" w arytmetyce modulo, która zawsze musi być zastosowana w przypadku czasu.

### Priorytet

Przy programowaniu w czasie rzeczywistym ważne jest pierwszeństwo zdarzeń. Ogólnie konstrukcja wyboru **ALT** może być różna przy zadeklarowaniu wyboru jednego z dwóch wejść pojawiających się jednocześnie. Dlatego konstrukcje **ALT** i **PAR**, które mają mieć pierwszeństwo w programach czasu rzeczywistego powinny być poprzedzone słowem kluczowym **PRI**.

## 7. Zakończanie programów równoległych.

Przy przebiegu procesów równoległych pojawia się problem ich zakończenia, gdyż na skutek ciągłego komunikowania się ich między sobą, zakończenie jednego, może spowodować, że inne będą stale się wykonywać lub też nastąpi zawieszenie innych procesów. Jeżeli procesy **PAR** wykonują się równolegle i dane są przesyłane między nimi na zasadzie "rurociagu", należałoby przyjąć sygnał kończący na wejściu, przysyłać go kolejno poprzez wszystkie procesy, a wreszcie zakończyć pracę, gdy sygnał ten dotrze do ostatniego procesu w ciągu. Inaczej procesy wykonujące się przy użyciu **ALT**, gdzie nie można przewidzieć, który z pojawiających się sygnałów zakończenia powinien być brany pod uwagę. Należałoby wtedy użyć **PRI PAR** w celu wskazania sygnału o najwyższym priorytecie.

## 8. Programy narzędziowe

Do pomocy w uruchamianiu programów w języku OCCAM służy zestaw narzędzi programowych tzw. *occam2 toolset*. Pozwala on redagować programy za pomocą konwencjonalnego edytora. Następnie są one kompilowane i linkowane za pomocą programów rezydujących w komputerze głównym lub i w głównej płycie transputerowej. Oddzielne narzędzie służy do produkowania kodu bootstrapa dla pojedynczych transputerów i dla sieci transputerowych w celu ładowania programu z uwzględnieniem połączeń transputerowych. Jest też cała gama pomocniczych programów narzędziowych i bibliotecznych, zawierająca również bibliotekę graficzną. Istnieje debugger sieciowy i symulator pozwalający testować programy bez sprzętu transputerowego.

Programy mogą być pisane również w języku C, Pascal lub Fortran i dołączane do innych modułów za pomocą specjalnych funkcji języka OCCAM.

Wybrane narzędzia języka OCCAM 2:

**occam** - kompilator (*cross-compiler*) języka OCCAM, który z kodu źródłowego tworzy modul wynikowy (do linkowania). Przy każdej kompilacji musi być wyspecyfikowany procesor docelowy;

**ilink** - program scalający moduly wynikowe i niezbędne moduly biblioteczne do postaci jednego programu wykonywalnego, zawierającego wszystkie odwołania zewnętrzne. Przeważnie jest to program przeznaczony na jeden procesor,

**iboot** - dodaje bootstrap do programu wynikowego. Taki ladowny program może być zaladowany do pojedynczego transputera i wykonany,

**iserver** - laduje programy do transputera lub sieci transputerów, a także określa środowisko, przez które program komunikuje się z komputerem głównym i systemem terminali,

**iconf** - buduje programy multitransputerowe na podstawie opisu konfiguracji i modułów wynikowych otrzymanych z programu ilink. Jako wynik jego działania może być zaladowana przez server kompletna sieć transputerowa,

**ilist** - generuje dokumentację modułów wynikowych i bibliotecznych;

**imakef** - automatyzuje generację modułów wykonywalnych ustalając, które części programu uległy zmianie od ostatniej kompilacji,

**idebug** - debugger analizuje stan zastopowanej sieci transputerowej,

**isism** - symulator T414,

**idump** - używany podczas uruchamiania do przechowania zawartości pierwszego transputera w sieci,

**iskip** - pozwala ladować program do sieci transputerów nie wykorzystując głównego transputera, który staje się jak gdyby przezroczysty dla danych przesyłanych między komputerem, programem i siecią transputerów,

**ilibr** - narzędzie tworzące pliki biblioteczne z różnych modułów wynikowych. Może generować takie pliki dla różnych procesorów docelowych. Tworzy on podstawowe moduly wykorzystywane przez ilink(linker).

OCCAM posiada również biblioteki zawierające programy pomocnicze, a więc biblioteki użytkownika zawierające najczęściej wykonywane procedury oraz biblioteki kompilatora, które są dołączane automatycznie przy kompilacji. Oto niektóre z nich:

**occam.lib** - biblioteka podstawowa zawierająca funkcje arytmetyki podwójnej długości, zmiennego przecinka, działań na bitach, instrukcji arytmetycznych itp.,

**maths.lib** - funkcje matematyczne,

**string.lib** - procedury przetwarzania ciągów,

**hostio.lib** - procedury dostępu do komputera głównego i systemu plików,

**msdos.lib** - procedury dostępu do funkcji DOS-u,

**streamio.lib** - procedury obsługi wc/wy,

**convert.lib** - procedury konwersji typów,

**xlink.lib** - procedury dostępu do połączeń zewnętrznych,

**graphics.lib** - biblioteka graficzna i inne.

Innym dostępnym narzędziem programowym jest system TDS (*Transputer Development System*), który umożliwia i ułatwia tworzenie programów współbieżnych na pojedyncze transputery i sieci transputerowe składające się z transputerów różnych typów.



## 9. Zakończenie

Ograniczenie mocy obliczeniowej, głównie związane z komunikacją i wymianą wyników obliczeń między procesorami, spowodowało tendencję do odchodzenia od architektur transputerowych na rzecz maszyn masywnie zrównoleglonych o większych mocach przetwarzania i pamięciach. Jednak firma INMOS, chcąc utrzymać się na rynku, zapewnia o rychłym wprowadzeniu drugiej generacji modułów transputerowych T9000 w technologii HTRAM (*High performance TRAnsputere Module*) charakteryzujących się wysoką gęstością upakowania elementów. Układ scalony T9000 zawiera 32-bitową jednostkę arytmetyczno-logiczną, 64-bitową jednostkę zmiennoprzecinkową, procesor komunikacyjny obsługujący cztery szerokopasmowe linie dwukierunkowej komunikacji z innymi transputerami, a także szerokopasmową linię do komunikacji z pamięcią RAM. Transputer T9000 pracuje z częstotliwością 50 MHz i adresuje pamięć zewnętrzną o pojemności do 4 GB. Łąca komunikacyjne pozwalają na transmisję z szybkością 80 MB/s.

Firma INMOS dąży również do otwarcia systemu poprzez oferowanie interfejsu między transputerami i innymi standardami jak MS DOS, Windows, X-Windows i komunikacji TCP/IP.

Postępuje również rozwój języka OCCAM, który będąc prostym i potężnym narzędziem projektowania równoległego i synchronizacji komunikacji pomiędzy procesami równoległymi nie spotkał się jednak z powszechną akceptacją, ponieważ mógł być stosowany tylko dla transputerów. Testowane są następne wersje, jak np. OCCAM 2.5, który posiada kompilator generujący kod 386 PC, a także inne narzędzia programowe - jak SPOC będący konwerterem OCCAM2 na język C i działający na stacjach SPARC pod systemem UNIX. Ostatnio zapowiadane jest również pojawienie się języka OCCAM3, który ma być zorientowany obiektowo.

Przyszłość pokaże, czy te nowości spowodują masowy zwrot w kierunku zastosowania transputerów.

## Literatura

- [1] OCCAM2 Reference Manual, Prentice Hall 1988
- [2] A tutorial introduction to OCCAM programming, Dick Pountain, David May, 1988
- [3] The transputer development and system datebook, INMOS 1989
- [4] The transputer applications notebook, Architecture and software, INMOS 1989
- [5] The transputer applications notebook, Systems and Performance, INMOS 1989
- [6] OCCAM2 toolset user manual, INMOS 1989
- [7] Graphics library for OCCAM2 toolset. KAT .I\_PL : 990
- [8] Zdzisław Szerbiński - Ocena efektywności i funkcjonowania komputerów o architekturze równoległej - Informatyka, z. 16, Gliwice 1993
- [9] WoTUG News for all users of the transputer, June 1994
- [10] TTC Transputing, July 1994
- [11] Introduction to concurrency using OCCAM, R.D. Dowsing





**ANDRZEJ KOJEMSKI  
LECH NAUMOWSKI  
TADEUSZ SINKIEWICZ**

**INSTYTUT MASZYN MATEMATYCZNYCH WARSZAWA**

## **Emuloanalizator EMA-51 i jego wykorzystanie do uruchamiania systemów mikroprocesorowych**

### **Emulator-Analyzer EMA-51 and its application for development the microprocessors systems**

#### **Streszczenie**

Emuloanalizator EMA-51 jest profesjonalnym narzędziem wspomagającym uruchamianie sprzętu i oprogramowania struktur mikroprocesorowych zawierających różne wersje mikroprocesorów rodziny 8051.

W skład tego urządzenia wchodzi wzajemnie zsynchronizowane: emulator układowy i analizator stanów logicznych. Pozwala to precyzyjnie rejestrować i analizować procesy zachodzące zarówno w mikroprocesorze jak i w sterowanym przez ten mikroprocesor obiekcie.

#### **Abstract**

Emulator-Analyzer EMA-51 provides the professional tool to support the development of e and software on the microprocessors' structures comprising various versions of the microprocessors of the 8051 family.

The apparatus consists of the mutually synchronized in circuit emulator and logic states' analyzer. That gives capability of the precise registration and analysis of the processes in the microprocessor and in the object controlling by this microprocessor.

#### **1. Wstęp**

Przy uruchamianiu złożonych urządzeń ze sterowaniem mikroprocesorowym, szczególnie w przypadku prototypowych struktur wieloprocessorowych, zachodzi konieczność rozwiązywania wielu problemów sprzętowo-programowych, trudnych do pokonania bez efektywnych narzędzi wspomagania.

Typowe systemy wspomagania uruchamiania urządzeń mikroprocesorowych są przeznaczone dla urządzeń jednoprocessorowych i nie zawierają środków do rozwiązywania zagadnień wzajemnej synchronizacji procesów i usuwania konfliktów czasowych sygnałów.

Prezentowany emuloanalizator EMA-51 stanowi kompleksowy zestaw sprzętowo-programowych narzędzi wspomagania do projektowania i uruchamiania w czasie rzeczywistym struktur jedno- i wieloprocesorowych zawierających mikroprocesory rodziny 8051, pozwalający uwzględniać rzeczywiste zależności czasowe występujące przy współpracy tych mikroprocesorów z obiektami o zróżnicowanych charakterystykach technicznych.

## 2. Koncepcja rozwiązania

Podstawowymi i najefektywniejszymi urządzeniami stosowanymi w procesie uruchamiania i testowaniu złożonych systemów mikroprocesorów są emulatory układowe mikroprocesorów i analizatory stanów logicznych. Między tymi urządzeniami uruchomieniowymi występują zasadnicze różnice. Analiza tych różnic prowadzi do wniosku, że urządzenia te wzajemnie się dopełniają, a ich jednoczesne użycie podczas uruchamiania umożliwia uzyskanie pełnej informacji o procesach zachodzących w uruchamianych systemach.

Emulatory układowe umożliwiają kontrolowanie i obserwację pracy mikroprocesorów w uruchamianym systemie. Umożliwiają także rozpoczęcie i zakończenie wykonywania wybranych fragmentów uruchamianego oprogramowania. Ponadto emulatory zapewniają dostęp do rejestrów mikroprocesorów oraz pamięci i portów uruchamianego systemu. Typowe emulatory układowe mają ograniczone możliwości rejestracji procesów zachodzących w uruchamianym urządzeniu poza mikroprocesorami. Rejestracja tych przebiegów, zdejmowanych za pomocą ograniczonej liczby sond, odbywa się wyłącznie w takt pracy magistrali mikroprocesora emulacyjnego, a rozpoczęcie i zatrzymanie rejestracji jest ograniczone własnościami układów punktów zatrzymania emulatorów, dla których wielkościami wejściowymi są sygnały magistrali procesora emulacyjnego.

W przeciwieństwie do emulatorów, analizatory stanów logicznych nie mają żadnych możliwości oddziaływania na przebieg pracy uruchamianych systemów, umożliwiają natomiast rejestrację w różnej skali czasowej przebiegów cyfrowych występujących w dowolnych punktach uruchamianych urządzeń. Zapamiętywanie tych przebiegów jest sterowane za pomocą układów synchronizacji, dla których wielkościami wejściowymi są również sygnały otrzymywane z dowolnych punktów uruchamianego systemu.

W uruchamianych systemach występuje wyraźny podział na mikroprocesor (lub grupę mikroprocesorów) z pamięciami i pozostałe układy logiczne, np. układy we/wy, układy bezpośrednio współdziałające ze sterowanym obiektem itp. Te dwie różne części systemu są zazwyczaj uruchamiane za pomocą różnych urządzeń: emulatora i analizatora. Jednocześnie działanie mikroprocesora jest ściśle związane z tym, co dzieje się w pozostałej części systemu. Z tego powodu tylko jednoczesne użycie emulatora i analizatora stanów podczas uruchamiania umożliwia uzyskanie pełnej wiedzy o procesach zachodzących w uruchamianych systemach. Celowa jest więc koordynacja działania emulatora i analizatora, np. poprzez oprogramowanie umożliwiające równoległe sterowanie tych urządzeń i obserwację wyników oraz sprzętową synchronizację ich pracy.

Zbadanie reakcji mikroprocesora na sygnały zewnętrzne i procesy zachodzące w układach logicznych uruchamianego systemu wymaga mechanizmów synchronizacji pracy układów wyzwalania analizatora i pamięci śladu emulatora. Także dla zbadania



reakcji systemu na wykonywane przez mikroprocesor oprogramowanie potrzebne są mechanizmy synchronizacji układów startu emulacji i punktów zatrzymania emulatora z układami wyzwalania analizatora.

Procesy zachodzące w układach uruchamianego systemu poza mikroprocesorem są związane przede wszystkim z otoczeniem uruchamianego systemu i z reguły przebiegają wolniej niż wykonywanie rozkazów programu przez mikroprocesor. W tych przypadkach do uzyskania pełnego obrazu pracy uruchamianego urządzenia potrzebny jest emulator i działający wolniej analizator. Czasami w celu znalezienia przyczyn błędnego działania uruchamianego urządzenia niezbędna jest rejestracja przebiegów logicznych z możliwie największą częstotliwością, umożliwiającą znalezienie wyścigów, hazardów i szpilkowych zakłóceń.

Niektórzy producenci sprzętu uruchomieniowego zaspokajają potrzeby związane z uruchamianiem złożonych systemów mikroprocesorowych. Na przykład Hewlett Packard produkuje system uruchomieniowy HP 64000, w którym uniwersalny emulator jest sprzężony z 16-kanalowym analizatorem stanów logicznych. W systemie HP 64000 układy wyzwalania analizatora współpracują z systemem punktów zatrzymania emulatora, umożliwiając korelację aktywności mikroprocesora z działaniem pozostałej logiki uruchamianych urządzeń.

Emuloanalizator EMA-51 opracowywany w IMM pozwala tworzyć systemy uruchomieniowe tego typu. Przy konstruowaniu emuloanalizatora EMA-51 wykorzystano opracowany wcześniej emulator układowy EM-51 i analizator stanów logicznych ASL-32. W dotychczasowym wykonaniu urządzenia te jako autonomiczne moduły systemu MSWP-92 są przeznaczone do samodzielnej pracy i nie jest możliwe koordynowanie ich działania.

Emuloanalizator EMA-51 jest efektywnym profesjonalnym zestawem wspomagającym uruchamianie sprzętu i oprogramowania systemów mikroprocesorowych zawierających mikrokomputery jednoukładowe rodziny 8051. W skład zestawu wchodzi analizator stanów logicznych ASL-32, wykonany w postaci karty rozszerzenia funkcjonalnego komputera personalnego IBM PC i emulator układowy EM-51, wykonany w postaci autonomicznego modułu dołączanego do mikrokomputera sterującego przez interfejs RS-232.

Oprogramowanie sterujące emuloanalizatorem EMA-51 działające pod systemem operacyjnym MS Windows jest przyjazne dla użytkownika. Oprogramowanie charakteryzuje wygodny dostęp do rozwijalnego menu za pomocą myszy i zawiera łatwo dostępny system pomocy (*help*).

Informacje dotyczące aktualnego stanu programu wykonywanego przez mikroprocesor i wartości wybranych sygnałów sterowanego obiektu są dostępne w odpowiednich oknach na ekranie monitora. Okna te mogą być przesuwane w dowolny obszar ekranu, a ich wymiary mogą być zmieniane zależnie od potrzeb.

Oprogramowanie wewnętrzne modułów emuloanalizatora daje możliwość programowego ustalania parametrów pracy, zapewnia szybką transmisję zbiorów reprezentujących zawartość pamięci emulacyjnej i efektywną interpretację zleceń wejściowych.

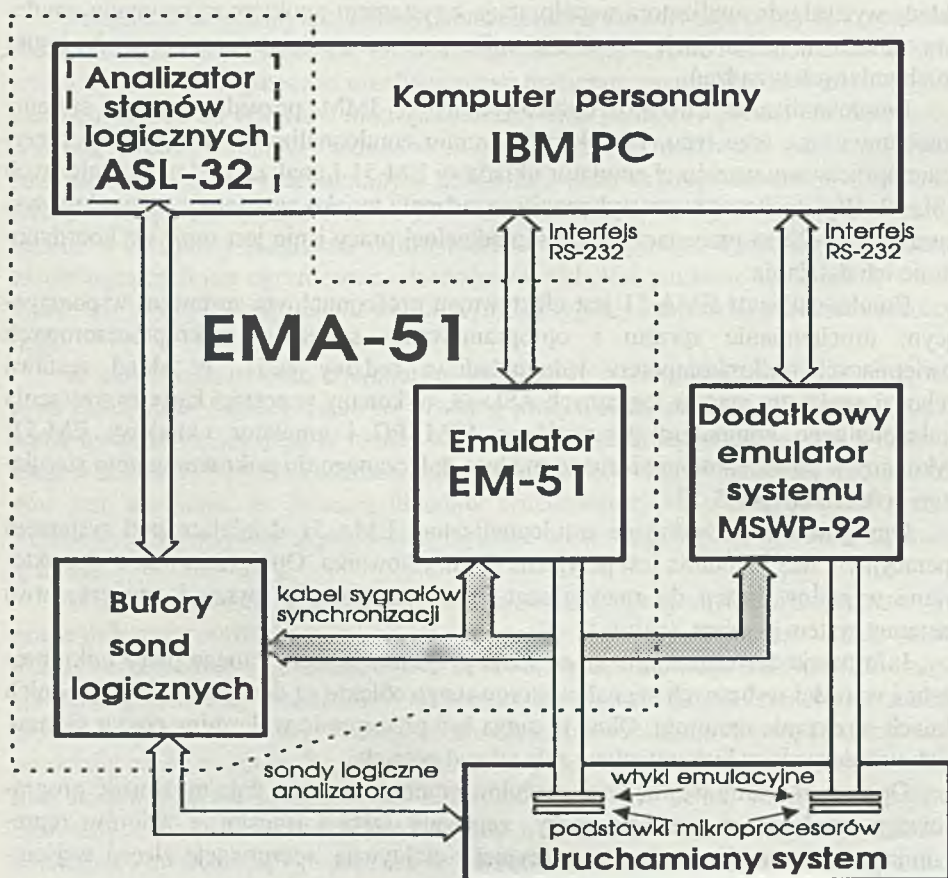
### 3. Struktura blokowa emuloanalizatora

Strukturę blokową emuloanalizatora z dołączonym do niego uruchamianym systemem i komputerem nadrzędnym przedstawia rys. 1.

Zmodyfikowany emulator EM-51 i analizator ASL-32 połączone są przewodami sygnałów synchronizacji oraz przewodem sygnału zegara. Dodatkowo w skład zestawu może wchodzić generator sygnału zegarowego dla emulatora i analizatora.

Pracą emuloanalizatora steruje komputer nadrzędny typu IBM PC. Emulator jest połączony z komputerem poprzez interfejs szeregowy RS-232, natomiast karta analizatora jest umieszczona w obudowie komputera i dołączona wprost do jego magistrali wewnętrznej.

W zestawie uruchomieniowym przeznaczonym do uruchamiania urządzeń wieloprocessorowych możliwe jest współdziałanie dwóch lub więcej emulatorów i analizatora stanów logicznych, połączonych przewodami sygnałów synchronizacji i zegarów.



Rys. 1. Struktura blokowa systemu zawierającego emuloanalizator EMA-51, zastosowany do uruchamiania systemu wieloprocessorowego.



#### 4. Charakterystyka emulatora

Jak wyżej wspomniano, emulatory układowe EM-51 są przeznaczone do uruchamiania i testowania sprzętu oraz oprogramowania systemów i urządzeń zawierających mikrokomputery jednoukładowe rodziny INTEL 8051 i pochodne.

Emulator zastępuje mikroprocesor w uruchamianym systemie w pełnym zakresie pod względem funkcjonalnym i wykonuje uruchamiane oprogramowanie bez żadnych ograniczeń związanych z prędkością, zakresem adresów pamięci i portów oraz dopuszczalnością stosu, rejestrów i przerwań.

Możliwa jest emulacja mikrokomputerów w typy Intel 8051/52, Oki 80154, Siemens 80515 w wykonaniach z wewnętrzną pamięcią programu lub bez oraz mikrokontrolerów Intel 8044 i Philips 80552/562 w wersji z zewnętrzną pamięcią programu. Planuje się przystosowanie emulatora EM-51 do emulacji innych mikroprocesorów firmy Philips, jak np. 80410/411, 80580, 80781/782, 80451 oraz mikroprocesorów innych firm stanowiących rozwinięcie mikroprocesora typu 8051.

Emulator jest dołączany do uruchamianego urządzenia za pomocą kabla emulacyjnego, którego wtyk jest umieszczany w gnieździe czerpanego mikroprocesora. Emulator działa tak jak rzeczywisty mikroprocesor wykonując z pełną prędkością wszystkie jego funkcje. Jest urządzeniem autonomicznym posiadającym własną obudowę i zasilacz, a dołączany jest do komputera nadrzędnego (typu IBM PC) za pomocą interfejsu szeregowego RS-232. Składa się z następujących modułów funkcjonalnych (rozmieszczonych na czterech pakietach):

- blok mikroprocesora sterującego z pamięcią programu sterującego i układem interfejsu szeregowego,
- blok mikroprocesora emulacyjnego z układem sterowania emulacją i pracą krokową,
- blok pamięci emulacyjnej zastępującej wewnętrzną (do 16 kB) i zewnętrzną (64 kB) pamięć programu oraz wewnętrzną (do 256 B) i zewnętrzną (64 kB) pamięć danych,
- blok punktów zatrzymania obejmujący trzy punkty zatrzymania dla stanów magistrali mikroprocesora emulacyjnego i dwa dla sygnałów zewnętrznych,
- blok pamięci śladu o pojemności 8..9.. słów 56-bitowych,
- blok układów synchronizacji zewnętrznej.

Sprzęt emulatora i oprogramowanie znajdujące się w pamięci stałej emulatora oraz oprogramowanie nadrzędne rezydujące w komputerze IBM PC realizują następujące funkcje:

- wyświetlanie repertuaru dostępnych poleceń (pomoc),
- zerowanie emulatora lub procesora emulacyjnego,
- określenie źródła zegara procesora emulacyjnego (wewnętrzny 16 MHz albo z uruchamianego urządzenia)
- emulację wewnętrznego pamięci kodu (4/8/16 kB), wewnętrznej pamięci danych (128/256 B), zewnętrznej pamięci kodu (64 kB) i zewnętrznej pamięci danych (64 kB),
- emulację pamięci zewnętrznych w blokach po 4 kB, z których każdy może być dołączony lub odłączony albo określony jako nicistniejący,

- asemblację liniową,
- deasemblację,
- wyświetlanie, modyfikowanie, wypełnianie, przenoszenie, przeszukiwanie, porównywanie zawartości pamięci,
- testowanie pamięci,
- zapis i odczyt portów we/wy,
- wyświetlanie i modyfikację zawartości rejestrów mikroprocesora emulacyjnego,
- blokowanie i odblokowywanie wybranych sygnałów sterujących mikroprocesora emulacyjnego,
- rozpoczęcie wykonywania programu uruchamianego urządzenia od zadanego adresu,
- ustawianie sprzętowych punktów zatrzymania emulacji:
  - czterech, dla których parametrami są adres, dana, status i ilość powtórzeń,
  - dwóch, dla których parametrami są sygnały zewnętrzne.
- tworzenie śladu wykonywanego programu z warunkowym rozpoczęciem albo zakończeniem śladowania,
- wyświetlanie i deasemblację zawartości pamięci śladu,
- pracę krokową w cyklach rozkazów mikroprocesora emulacyjnego,
- ładowanie zbiorów z pamięci dyskowej komputera nadrzędnego do pamięci emulacyjnej oraz zapamiętywanie zawartości tej pamięci w pamięci dyskowej komputera sterującego z prędkością do 11 kB/s (115400 bodów),
- działanie na tablicy symboli i posługiwanie się symbolami w poleceniach dla emulatora,
- tworzenie zbiorów poleceń, które mogą zawierać pętle działań warunkowych, umożliwiających automatyczne testowanie uruchamianego urządzenia,
- zapamiętywanie przebiegu sesji uruchomieniowej,
- wydruk bieżącej informacji z ekranu monitora,
- zobrazowanie zawartości pamięci śladu w postaci diagramów stanów logicznych,
- analizę częstości wywoływania i czasu wykonywania wybranych modułów podczas emulacji,
- synchronizację pracy emulatora z analizatorem stanów logicznych obejmującą:
  - generację sygnału synchronizacji wyzwalania analizatora w momentach zatrzymania emulacji albo zadziałania układów punktów zatrzymania emulatora,
  - zatrzymanie emulacji lub rozpoczęcie/zatrzymanie działania pamięci śladu na sygnał synchronizacji z analizatora stanów logicznych,
  - generację sygnału zegarowego dla analizatora.

## 5. Charakterystyka analizatora stanów logicznych

Analizator stanów logicznych ASL-32 jest przeznaczony do śledzenia sygnałów cyfrowych. Zapamiętuje 16 lub 32 stany sygnałów próbkowanych z częstotliwością odpowiednio 100 i 50 MHz. Wykonany w postaci karty umieszczanej w komputerze typu IBM PC ma dwa zespoły sond logicznych połączone z nią za pomocą płaskich kabli.



## Podstawowe dane techniczne analizatora:

- liczba kanałów pomiarowych (sond):
  - przy zegarze wewnętrznym o częstotliwości 100 MHz 16
  - przy zegarze wewnętrznym o częstotliwości od 100 Hz do 50 MHz 16 lub 32
  - przy zegarze zewnętrznym o częstotliwości do 50 MHz 16 lub 32
- pojemność pamięci śladu:
  - przy 16 kanałach (max. 100 MHz) 16 kb/kanał
  - przy 32 kanałach (max. 50 MHz) 8 kb/kanał
- źródło zegara i częstotliwość próbkowania:
  - zegar wewnętrzny przy 16 sondach 100 Hz - 100 MHz
  - zegar wewnętrzny przy 32 kanałach 100 Hz - 50 MHz
  - zegar zewnętrzny 0 Hz - 50 MHz
- napięcie wejściowe:
  - maksymalne -15 V do + 15 V
  - zalecane -7 V do +7 V
- progi próbkowania:
  - poziom sygnałów TTL 1,4 V
  - poziom sygnałów ECL -1,3 V
  - programowany w zakresie -7 V do +7 V ze skokiem 0,2 V
- wyzwianie:
  - programowanym słowem wyzwajającym,
  - zewnętrznym impulsem wyzwajającym generowanym przez emulator,
  - generacja sygnału synchronizacji dla emulatora w momencie wyzwolenia wewnętrznego
- programowane opóźnienie zakończenia rejestracji danych względem momentu wyzwolenia w zakresie 0 - 16384 cykli ze skokiem 256

Program sterujący analizatorem daje możliwość wyboru sposobu wyświetlania zarejestrowanych przebiegów (graficznie albo kody ASCII, dziesiętnie, heksadecymalnie lub binarnie) oraz ilości wyświetlanych przebiegów. Możliwe jest wyświetlanie całości albo dowolnych fragmentów zarejestrowanych przebiegów z różnymi skalami czasowymi i przewijaniem ekranu. Dwa kusory ułatwiają analizę czasową przebiegów. Zawartość pamięci analizatora oraz parametry pomiaru mogą być zapisywane w postaci zbiorów dyskowych.

## 6. Współdziałanie modułów analizatora i emulatora

Współdziałanie emulatora układowego EM-51 z analizatorem stanów logicznych ASL-32 jest możliwe dzięki istnieniu w tych urządzeniach układów synchronizacji zewnętrznej. Układy te spełniają dwa zadania:

- generują sygnały dla synchronizacji współpracującego urządzenia,
- włączają zaprogramowane funkcje po odebraniu sygnału synchronizacji od współpracującego urządzenia.

Emulator EM-51 generuje sygnały synchronizacji zewnętrznej spowodowane przez:

- zadziałanie układów punktów zatrzymania emulatora w momencie stwierdzenia zgodności stanów na magistrali mikroprocesora emulacyjnego z zaprogramowanym warunkiem,
- uruchomienie albo zatrzymanie pamięci śladu,
- zatrzymanie emulacji.

Układy synchronizacji zewnętrznej w emulatorach po odebraniu sygnałów synchronizacji mogą spowodować:

- zatrzymanie emulacji,
- rozpoczęcie/zatrzymanie działania pamięci śladu.

Dodatkowo układy synchronizacji umożliwiają nadawanie sygnału zegarowego mikroprocesora emulacyjnego do analizatora stanów logicznych i innych urządzeń uruchomieniowych. W tym przypadku źródłem sygnałów zegarowych może być zegar wewnętrzny emulatora albo układ uruchamiany. Możliwe jest także dostarczanie sygnałów zegarowych dla mikroprocesora emulacyjnego i analizatora z niezależnego zewnętrznego programowanego generatora.

Emulatory mają ponadto możliwość wymiany sygnałów synchronizacji rozpoczęcia emulacji, niezbędnych podczas uruchamiania systemów wielomikroprocesorowych uruchamianych za pomocą dwóch lub więcej emulatorów. W takich sytuacjach z reguły wymagane jest zapewnienie jednoczesnego startu emulacji wszystkich emulowanych mikroprocesorów.

Analizator stanów logicznych ASL-32 może pracować z wewnętrzną albo zewnętrzną synchronizacją wyzwalania rejestracji badanych sygnałów. Zaprogramowany na pracę z wyzwalaniem wewnętrznym może generować wyjściowy sygnał synchronizacji. Sygnał ten jest nadawany w momencie zadziałania układu wyzwalania po stwierdzeniu zgodności stanów na sondach logicznych z zaprogramowanym warunkiem i może być użyty do sterowania pracą pamięci śladu w emulatorach i/lub zatrzymania emulacji.

Po zaprogramowaniu analizatora na pracę z wyzwalaniem zewnętrznym rejestracja przebiegów z sond logicznych jest sterowana przez wejściowy sygnał synchronizacji zewnętrznej generowany przez emulator. Dodatkowo analizator jest wyposażony w wejście sygnałów zegarowych, dzięki któremu jest możliwe zsynchronizowanie procesu próbkowania w analizatorze z zegarem mikroprocesora emulacyjnego współpracującego emulatora.

Stanowisko uruchomieniowe przeznaczone do uruchamiania złożonych systemów mikroprocesorowych, pokazane na rys. 1, składa się z:

- uruchamianego systemu,
- emulatora lub wielu emulatorów,
- analizatora stanów logicznych wraz z buforami sond logicznych,
- komputera nadrzędnego typu IBM PC.

W przypadku uruchamiania systemu z wieloma mikroprocesorami do sterującego mikrokomputera można dołączyć oprócz emuloanalizatora dodatkowe emulatory układowe. Karta analizatora stanów logicznych umieszczona wewnątrz komputera dołączona jest bezpośrednio do magistrali AT-bus (ISA), a emulatory są dołączone przez złącza interfejsu szeregowego RS-232. Emulator lub emulatory i analizator są



kontrolowane za pomocą programów sterujących działających współbieżnie w środowisku Windows. Umożliwia to szybkie sekwencyjne wprowadzanie poleceń dla poszczególnych urządzeń uruchomieniowych i obserwację wyników ich działania.

Podczas uruchamiania moduły sond logicznych analizatora, tak jak i emulator (emulatory), są umieszczone w pobliżu uruchamianego układu. Wtyki kabli emulacyjnych emulatorów są włożone w podstawki odpowiednich mikroprocesorów. Sondy logiczne emulatorów oraz sondy analizatora są dołączone do wybranych punktów uruchamianego urządzenia. Połączenia te umożliwiają uzyskanie bogatej informacji o pracy mikroprocesorów i innych układów w systemie. Emulator i analizator stanów logicznych wchodzące w skład systemu uruchomieniowego i stanowiące emuloanalizator są połączone przewodami sygnałów synchronizacji. Użytkownik ma możliwość dołączenia wejść i wyjść synchronizacji innych emulatorów (tj. nie wchodzących w skład wyróżnionego funkcjonalnie emuloanalizatora) z wejściami lub wyjściami synchronizacji w modułach emuloanalizatora. Połączenia te umożliwiają synchronizację działania urządzeń tworzących złożony system uruchomieniowy.

Możliwe jest także uzyskanie synchronizacji momentów rejestracji stanów magistrali mikroprocesora emulacyjnego w pamięci śladu emulatora oraz próbkowania i rejestracji przebiegów logicznych w pamięci analizatora. W tym celu emulator i analizator mogą być połączone kablem przesyłającym sygnały zegarowe mikroprocesora emulacyjnego do analizatora. Możliwe jest także dostarczenie zsynchronizowanych sygnałów zegarowych dla emulatora i analizatora z dodatkowego programowanego generatora.

Każdy krok procesu uruchomieniowego po odpowiednim połączeniu sond logicznych i przewodów synchronizacji/wyzwalania obejmuje ustawienie warunków pracy analizatora i emulatora, a następnie uruchomienie emulacji. Zapis danych w pamięci analizatora i w pamięci śladu emulatora oraz ewentualne zatrzymanie emulacji następuje po zadziałaniu układu wyzwalania analizatora lub układu punktów zatrzymania emulatora, w zależności od uprzednio ustawionych warunków. Po zgromadzeniu danych w analizatorze i pamięci śladu emulatora, możliwy jest ich odczyt i analiza.

## 7. Oprogramowanie sterujące

Oprogramowanie sterujące emuloanalizatora EMA-51 jest przystosowane do pracy w środowisku Windows 3.1 lub nowszym i zawiera trzy składniki:

- oprogramowanie sterujące analizatora ASL-32,
- oprogramowanie sterujące emulatora EM-51,
- oprogramowanie sterujące współdziałaniem obu wskazanych modułów.

Oprogramowanie sterujące analizatora ASL-32 jest autonomicznie działającym programem, co oznacza, że oprogramowanie to można wykorzystywać przy samodzielnej pracy analizatora, bez uwzględniania współdziałania z emulatorem. W takim przypadku nie są wykorzystywane specjalnie wbudowane funkcje tego oprogramowania stosowane przy pracy emuloanalizatora.

Program sterujący analizatora stanów logicznych ASL-32 w sposób dogodny dla użytkownika pozwala wprowadzać parametry pomiarów, wizualizować wyniki na ekranie w postaci wykresów czasowych lub kolumn znaków alfanumerycznych oraz utrzymywać wyniki w postaci wydruku lub zbioru dyskowego. Zbiór dyskowy zawierający

ustawione przez użytkownika parametry, a także zawartość odpowiadającej im pamięci śladu, mogą być następnie wykorzystywane jako wzorce do wielokrotnego testowania takich samych układów użytkownika. Umożliwia to opcja automatycznego porównania zawartości zbioru dyskowego z ustawionymi aktualnie parametrami i zarejestrowanymi w pamięci śladu wynikami.

Program sterujący analizatora jest programem typu okienkowego z własną nazwą, wyposażonym w menu służące do wybierania poleceń. Realizowane funkcje są podobne do funkcji uprzednio opracowanego w IMM analizatora stanów logicznych ASL-24.

Badanie przebiegów czasowych przy użyciu analizatora poprzedza zadanie parametrów pomiarów. Parametry mogą być wprowadzane przez użytkownika ręcznie lub wczytane ze zbioru dyskowego.

W okienku programu są automatycznie wyświetlane ustawiane przez użytkownika parametry pracy analizatora, takie jak: dane zegara, poziom wyzwalania, słowo wyzwalające i opóźnienie.

Odpowiednie zaprogramowanie warunków pracy analizatora umożliwia automatyczny zapis przebiegów czasowych w pamięci śladu analizatora w przypadku spełnienia warunków wyzwolenia.

Zarejestrowane przez analizator przebiegi zapamiętane w zbiorze dyskowym mogą posłużyć w późniejszym okresie do analizy, porównań itp.

Oprogramowanie sterujące emulatora EM-51 jest autonomicznie działającym programem, co oznacza, że oprogramowanie to można wykorzystywać przy samodzielnej pracy emulatora, bez uwzględniania współdziałania z analizatorem. Przy korzystaniu wyłącznie z emulatora nie są wykorzystywane specjalnie wbudowane funkcje tego oprogramowania stosowane w przypadku pracy emuloanalizatora.

Program sterujący emulatora układowego EM-51 w sposób dogodny dla użytkownika pozwala wprowadzać polecenia umożliwiające badanie stanu uruchamianego obiektu (mikroprocesora, pamięci itp.), śledzenie oraz zapisywanie przebiegu programu w pamięci śladu oraz utrwalanie wyników w postaci wydruku lub zapisu do zbioru dyskowego.

Program ten jest programem typu okienkowego z własną nazwą, wyposażonym w menu służące do wybierania poleceń. Realizowane funkcje są podobne do funkcji odpowiadających uprzednio opracowanemu emulatorowi EM-51, z tym że ze względu na udogodnienia jakie daje środowisko Windows, zrezygnowano ze stosowania jako głównego sposobu wprowadzania poleceń wypisywanie ich w linii wejściowej wraz ze szczegółowym wypisywaniem argumentów. Zostało to zastąpione wybieraniem poleceń z menu i określaniem argumentów na wyświetlanych panelach dialogowych.

Przyjęto też, że interesujące dla użytkownika informacje, tego typu jak zawartości pamięci, stany rejestrów procesora i zawartość stosu pamięci są wyświetlane w oddzielnych oknach, które użytkownik może wywoływać lub związać w zależności od bieżących potrzeb.

Oprogramowanie sterujące współdziałaniem modułu analizatora i modułu emulatora także pracuje w środowisku Windows. Program ten jest stosowany tylko w przypadku wykorzystywania przez użytkownika współdziałania obu omawianych modułów.

Praca programu sterującego współdziałaniem modułów składowych emuloanalizatora polega głównie na ustalaniu określonej zawartości zbioru definiującego warunki współdziałania obu modułów. Dane te są określane przez użytkownika przez



wypełnianie odpowiedniej planszy dialogowej pokazującej się w postaci formularza w oknie programu sterującego.

Programy sterujące modułami emulatora i analizatora wykorzystują zapisane w powyższym zbiorze wartości do ustalenia warunków swego działania w momencie ich wywoływania. Możliwe jest też wymuszanie odczytu tych wartości przez wywołane już programy sterujące poszczególnych modułów, za pomocą odpowiednich poleceń z "wnętrza" tych programów.

## 8. Przykładowe sposoby wykorzystania emuloanalizatora

Odpowiednio zaprojektowane wejścia i wyjścia modułu emulatora oraz modułu analizatora służą do zapewnienia synchronizacji czasowej pracy obu tych modułów. Wszystkie te wejścia i wyjścia w obu modułach są sterowane programowo. Oznacza to, że z poziomu programu użytkownik może zdecydować, czy dana końcówka złącza synchronizacji w rozpatrywanym module odpowiada wyjściu lub też wejściu oraz jaki sygnał wyjściowy pojawia się na tej końcówce (gdy mamy do czynienia z wyjściem) lub jak jest wykorzystywany otrzymywany sygnał (gdy mamy do czynienia z wejściem). Umożliwia to trwale standardowe połączenie obu tych modułów kablem synchronizacji i wyznaczanie sposobu ich współpracy przez odpowiednieysterowanie programowe.

Należy zaznaczyć, że praktycznie tylko jeden z dwóch modułów może być nadawcą sygnału synchronizującego rozpoczęcie pracy drugiego modułu emuloanalizatora.

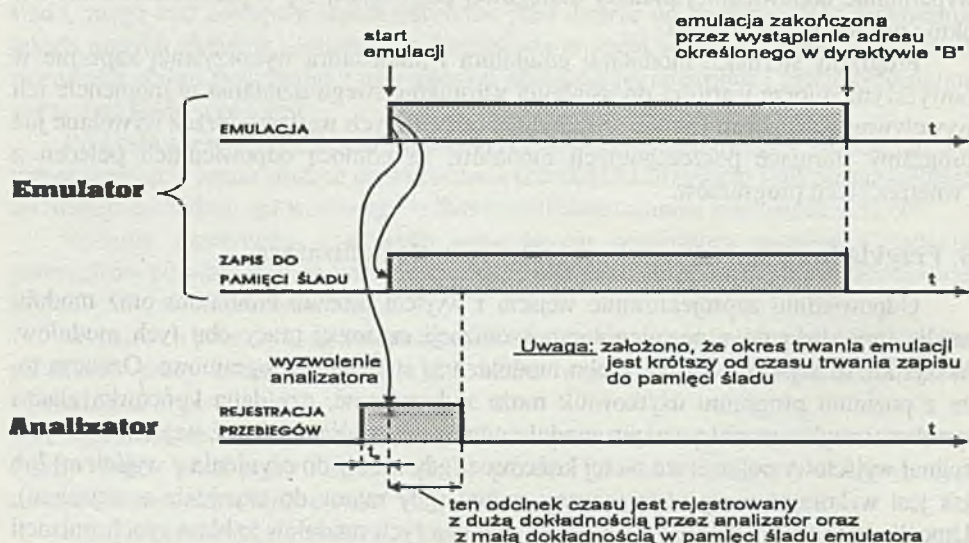
Jako pierwszy rozpatrzmy przypadek, w którym emulator wyzwala analizator. Ze względu na to, że analizator potrafi rejestrować przebiegi sygnałów z dużo większą częstotliwością niż odbywa się rejestracja przebiegów w pamięci śladu emulatora, analizator może być w tym przypadku z powodzeniem wykorzystywany jako "lupa czasowa" umożliwiająca użytkownikowi wniknięcie w szczegóły zmian interesujących go sygnałów. W tym celu użytkownik może ustalić warunki współpracy emulatora z analizatorem w ten sposób, że:

a) sygnał startu emulacji powoduje jednoczesne wyzwolenie analizatora; wykresy czasowe ilustrujące tę sytuację przedstawiono na rys. 2.,

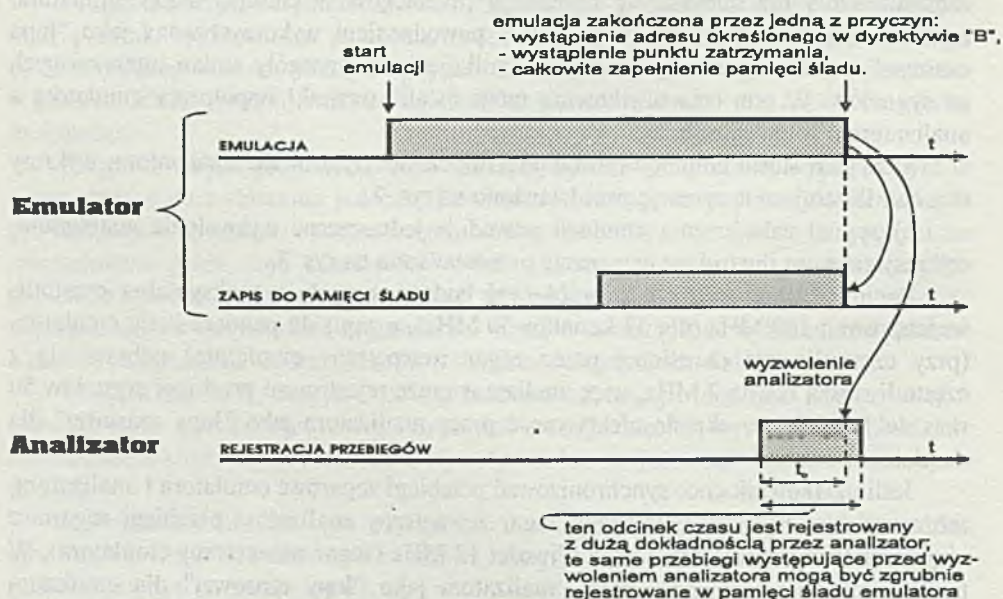
b) sygnał zakończenia emulacji powoduje jednoczesne wyzwolenie analizatora; wykresy czasowe ilustrujące tę sytuację przedstawiono na rys. 3.

Ponieważ analizator może próbować badane sygnały z maksymalną częstotliwością równą 100 MHz (dla 32 kanałów 50 MHz), a zapis do pamięci śladu emulatora (przy częstotliwości określonej przez zegar wewnętrzny emulatora) odbywa się z częstotliwością równą 2 MHz, więc analizator może rejestrować przebiegi sygnałów 50 razy dokładniej i to określa efektywność pracy analizatora jako "lupy czasowej" dla emulatora.

Jeśli użytkownik chce synchronizować przebiegi zegarowe emulatora i analizatora, może wówczas wykorzystywać jako zegar zewnętrzny analizatora przebiegi zegarowe wyprowadzone z emulatora o częstotliwości 12 MHz (zegar wewnętrzny emulatora). W tym przypadku efektywność pracy analizatora jako "lupy czasowej" dla emulatora maleje i gęstość zapisu dla analizatora jest 6 razy większa od zapisu w pamięci śladu emulatora.



Rys. 2. Harmonogramy współdziałania modułów emuloanalizatora. Start emulacji w emulatorze wyzwala analizator. Analizator pracuje jako "lupa czasowa".



Rys. 3. Harmonogramy współdziałania modułów emuloanalizatora. Zakończenie emulacji wyzwala analizator. Analizator pracuje jako "lupa czasowa".



Na rys. 2 i 3 wartość odcinka czasu oznaczona jako  $t_0$  może zmieniać się od zera do wartości maksymalnej, wynikającej z ustawienia parametrów pracy analizatora. Dla przypadku przedstawionego na rys. 2 odcinek czasu  $t_0$  odpowiada stanowi sygnałów przed rozpoczęciem emulacji i dlatego powinien być ustalany w pobliżu wartości najmniejszych. Dla przypadku przedstawionego na rys. 3 odcinek czasu  $t_0$  odpowiada stanowi sygnałów rejestrowanych jednocześnie przez analizator i w pamięci śladu emulatora, i dlatego powinien być ustalany w pobliżu wartości największych.

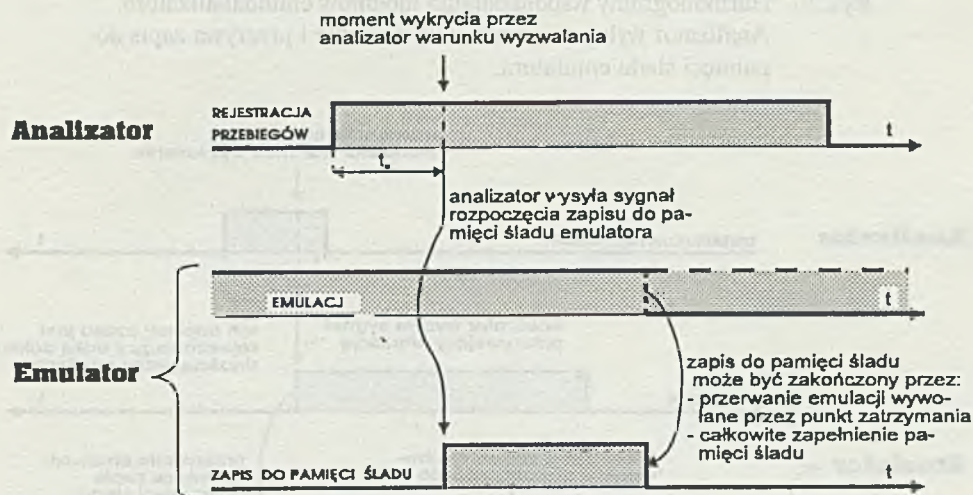
Jako drugi rozpatrzmy przypadek, w którym analizator wyzwała emulator i pracuje z częstotliwością rejestracji stanu sygnałów mniejszą niż częstotliwość zapisu do pamięci śladu emulatora.

Ponieważ analizator potrafi rejestrować przebiegi sygnałów z dużo mniejszą częstotliwością, niż odbywa się rejestracja przebiegów w pamięci śladu emulatora, analizator może być w tym przypadku z powodzeniem wykorzystywany jako nadrzędny układ sterujący z zewnątrz pracą emulatora. W szczególności analizator może być wykorzystany do:

a) rozpoczęcia zapisu do pamięci śladu emulatora; wykresy czasowe ilustrujące tę sytuację przedstawiono na rys. 4.,

b) przerywania zapisu do pamięci śladu emulatora; wykresy czasowe ilustrujące tę sytuację przedstawiono na rys. 5.

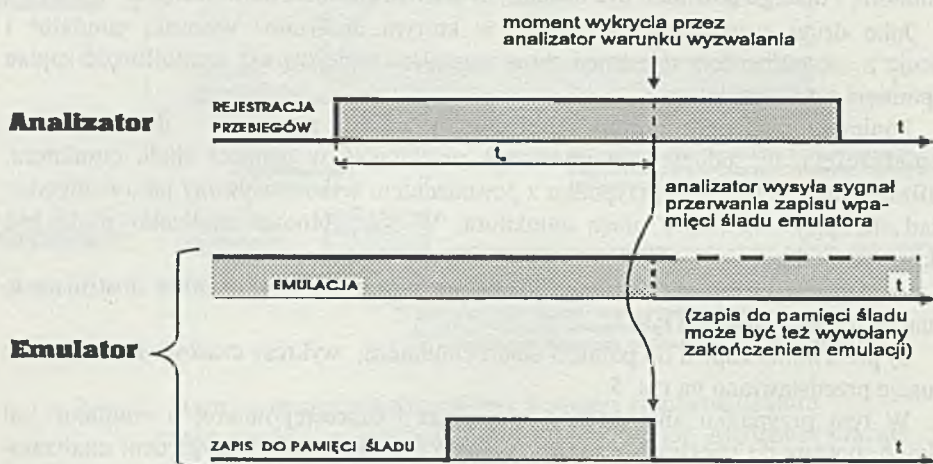
W tym przypadku analizator działa w skali czasowej makro, a emulator jest wykorzystywany do rejestracji i analizy sygnałów w skali mikro (względem analizatora).



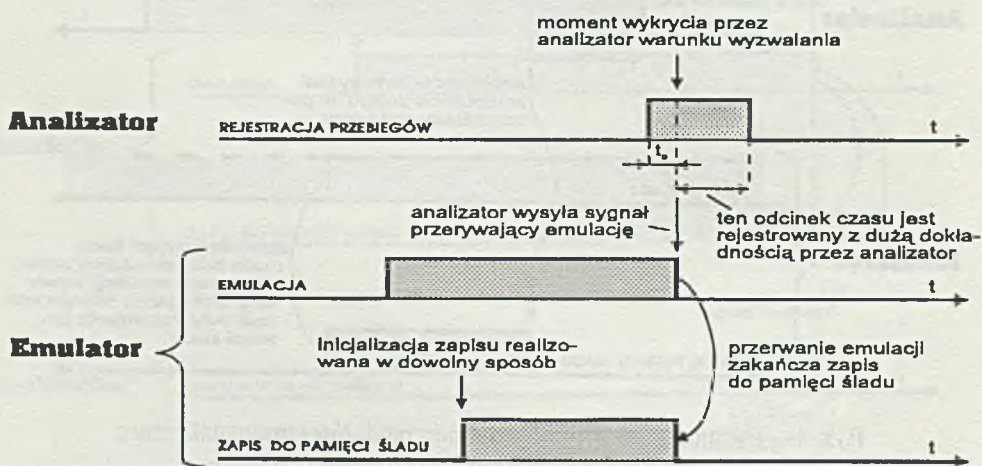
Rys. 4. Harmonogramy współdziałania modułów emuloanalizatora. Analizator wykrywa warunek wyzwania i wysyła sygnał powodujący rozpoczęcie zapisu do pamięci śladu emulatora.

Na rys. 4 i 5 wartość odcinka czasu oznaczona jako  $t_0$  może zmieniać się od zera do wartości maksymalnej, wynikającej z ustawienia parametrów pracy analizatora. Dla przypadku przedstawionego na rys. 4 odcinek czasu  $t_0$  odpowiada rejestracji sygnałów przez analizator przed rozpoczęciem zapisu do pamięci śladu emulatora. Wartość tego

odcinka czasu powinna być odpowiednio dobrana przez użytkownika w zależności od aktualnych potrzeb. Dla przypadku przedstawionego na rys. 5 odcinek czasu  $t_0$  odpowiada stanowi sygnałów rejestrowanych jednocześnie przez analizator i w pamięci śladu emulatora oraz ewentualnie rejestracji przebiegów sygnałów przez analizator przed rozpoczęciem zapisu do pamięci śladu emulatora. Powinien on być ustalany przez użytkownika zgodnie z aktualnymi potrzebami.



Rys. 5. Harmonogramy współdziałania modułów emuloanalizatora. Analizator wykrywa warunek wyzwania i przerywa zapis do pamięci śladu emulatora.



Rys. 6. Harmonogramy współdziałania modułów emuloanalizatora. Analizator wykrywa warunek wyzwania i wysyła sygnał powodujący przerwanie emulacji. Jednocześnie zostaje przerwany zapis do pamięci śladu emulatora. Analizator pracujący z dużo większą częstotliwością niż emulator działa jako "lupa czasowa" dla przebiegów rejestrowanych w pamięci śladu.



Jako trzeci przypadek rozpatrzmy sytuację, w której analizator wyzwała emulator i pracuje z częstotliwością rejestracji stanu sygnałów większą niż częstotliwość zapisu do pamięci śladu emulatora. Interesujący jest taki sposób połączenie analizatora z emulatorem, w którym analizator po wykryciu zaprogramowanych warunków wyzwalania rejestracji wysyła do emulatora sygnał zatrzymania emulacji. Harmonogram przebiegów czasowych dla tego przypadku przedstawia rys. 6. Analizator może być traktowany jako układ realizujący "lupe czasową" dla przebiegów rejestrowanych w pamięci śladu emulatora, jeśli przed przerwaniem emulacji w emulatorze był uruchomiony zapis do pamięci śladu. Uwagi dotyczące synchronizacji pracy zegarów w analizatorze i emulatorze podane poprzednio są aktualne również dla tego przypadku.





**JAN RYŻKO****INSTYTUT MASZYN MATEMATYCZNYCH WARSZAWA**

## **Rynek informatyczny, a zwłaszcza komputerów osobistych na świecie i w Polsce**

**The information technology market, especially for personal  
computers in the world and Poland.**

### **Streszczenie**

W pracy przytoczono obraz rynku informatycznego na początku lat dziewięćdziesiątych w świecie. Przytoczono wiele tabel rankingowych. Omówiono poszczególne dziedziny, a w szczególności rynek komputerów osobistych. Porównano różne procesory stosowane obecnie w tych systemach. Podano też główne firmy informatyczne w Polsce. Omówiono aktualne tendencje rynkowe i perspektywy na najbliższą przyszłość.

### **Abstract**

In the paper the image of information technology market in the beginning of ninty decade is presented. Many ranking tables are presented. Different divisions of the market are described, especially personal computers' market. Different microprocessors used in these systems are compared. The main computer companies and perspectives to nearest future are described.

Rynek informatyczny stanowi poważny dział rynku elektronicznego i jego udział w tym ostatnim ciągle wzrasta. Do niedawna tempo rozwoju rynku informatycznego wynosiło kilkanaście procent rocznie, ale w ostatnich latach takich wskaźników nie daje się utrzymać. Wśród materiałów opisujących sytuację na tym rynku największą popularnością cieszy się publikacja ogłaszana w czerwcu każdego roku przez czasopismo Datamation.

Tegoroczna publikacja [1], poza standardowymi listami pierwszych 100 firm w Ameryce Północnej i na świecie, przynosi wyniki ankiety na temat kryteriów wyboru dostawców techniki informatycznej przez klientów oraz 10 głównych firm w dziewięciu wybranych dziedzinach.

Przedstawiamy naszym czytelnikom główne fragmenty tego raportu, ograniczając się do pierwszych części publikowanych tabel, niekiedy uzupełniając je własną statystyką (np. zestawienie owych dziewięciu dziedzin). Wspominając swe pierwsze

raporty, autorzy ww. publikacji podkreślają, że wówczas brak było standaryzacji i odbiorca musiał bardzo starannie dobierać dostawcę, gdyż zły wybór mógł prowadzić do ogromnych strat ze względu na niewymiennność sprzętu oraz bezużyteczność opracowanego oprogramowania. Dziś nie ma już takiego niebezpieczeństwa, ale problem staje się bardziej złożony i należy więcej wiedzieć o wyrobach, które powinny być oferowane przez wielu dostawców. Dlatego zapytano ponad 300 użytkowników, czym kierują się przy wyborze dostawcy, a następnie jakie firmy spełniają te wymagania. Okazało się, że w 10-punktowej skali stawiają oni na pierwszym miejscu jakość i niezawodność wyrobu (9,6 pkt), potem jego sprawność (9,2), jakość usług (8,8), a dopiero później wiarygodność dostawcy (8,2), stosunek wydajności do ceny (8,1), łatwość zawierania transakcji z dostawcą (7,7), spełnianie przez dostawcę standardów branżowych (7,6), perspektywiczność (7,4) i wreszcie na końcu stabilność finansową dostawcy (7). Stosując te kryteria do 31 firm amerykańskich uzyskano tabelę, z której przytaczamy 10 pierwszych pozycji.

Tabela 1

Firma	wyd.	jak.	usl.	wiar.	std.	wd/c	fin.	tr.	prs.	kom
Microsoft	7,8	7,5	6,9	7,0	7,2	7,7	8,4	7,4	7,3	7,5
HP	7,5	7,8	7,2	7,5	7,3	6,8	7,7	7,0	7,1	7,3
Compaq	7,8	8,0	7,0	7,8	7,4	6,9	7,1	6,8	6,8	7,3
Sun	7,8	7,8	7,1	7,2	7,0	6,8	7,1	6,7	6,8	7,1
Novell	7,3	7,3	6,8	7,1	6,9	6,5	7,6	6,6	7,0	7,0
IBM	7,4	7,7	7,4	7,4	6,7	6,0	7,2	6,7	6,5	7,0
AT&T	6,9	7,0	6,6	7,1	6,9	6,1	7,7	6,6	6,5	6,8
Dell	6,8	7,0	6,8	7,0	7,1	7,1	6,0	7,1	6,6	6,8
Digital	7,0	7,1	6,9	7,0	6,7	6,4	6,4	6,6	6,7	6,8
Seagate	6,9	6,8	6,2	6,7	7,3	7,0	6,4	6,6	6,8	6,7

gdzie kolejne skróty oznaczają: wyd.- wydajność wyrobu, jak. - jakość i niezawodność wyrobu, usl. - jakość usług, wiar. - wiarygodność dostawcy, std. - zgodność ze standardami branżowymi, wd/c - stosunek wydajności do ceny, fin. - stabilność finansową dostawcy, tr. - łatwość współpracy z dostawcą, prs. - otwartość na przyszłościowe strategie i plany, kom. - współczynnik będący kombinacją wszystkich pozostałych.

Ostatnie miejsce w pełnej tabeli zajmuje Computervision z kombinowaną punktacją 5,2, a poprzedzają ją Wang i Tandem (po 5,4). Firmy o wysokim współczynniku kombinowanym zajmowały wysokie pozycje na listach rankingowych i wykazywały tendencje do poprawiania tych pozycji. Tabela 2 przedstawia 50 czołowych światowych firm informatycznych zestawionych według przychodów z techniki informatycznej w 1993 roku w mln dolarów USA. Podane tu są również przychody w roku 1992 i ich procentowa zmiana.



Przychody (w mln USD) największych firm informatycznych na świecie.

Tabela 2

1993	1992	Firma	Przychody w 1993 r.	Przychody w 1992 r.	zmiana [%]
1	1	IBM	62716	64520	-2,8
2	2	Fujitsu	21871,9	20142,2	8,6
3	3	NEC	16674,8	15395	8,3
4	4	HP	15600	12688	23,0
5	5	Digital	13637	14162	-3,7
6	6	Hitachi	12629,1	11352	11,2
7	7	AT&T	9860	10450	-5,6
8	10	Toshiba	8819,7	7448,7	18,4
9	15	EDS	8507,3	8218,9	3,5
10	11	Apple	7900	7173,7	10,1
11	8	Siemens Nixdorf	7225,5	8345,1	-13,4
12	9	Unisys	7200,5	7832	-8,1
13	18	Compaq	7200	4100	75,6
14	12	Olivetti	5070,2	5762	-12,0
15	14	Matsushita	5050,7	5060,8	-0,2
16	16	Canon	5033	4633,6	8,6
17	13	Groupe Bull	5000	5715,1	-12,5
18	19	Sun Electronics	4493	3832	17,2
19	20	Microsoft	4110	3253	26,3
20	17	ICL	3915,7	4354,8	-10,1
21	21	NTT	3905,6	3119,5	25,2
22	23	Xerox	3330	3016	10,4
23	22	Seagate	3114	3079,4	1,1
24	26	Mitsubishi	3042,5	2589,4	17,5
25	24	Andersen Consulting	2879,3	2720	5,7
26	36	Dell	2870	2013,9	42,5
27	25	Nihon Unisys	2697,8	2646,1	2,0
28	28	CSC	2502	2474,2	1,1
29	34	ADP	2339,1	2075	12,7
30	33	Ricoh	2290	2124,6	7,8
31	37	TRW	2251	1855	21,3
32	29	Nothern Telecom	2250	2100	7,1
33	30	OKI	2245,6	2289,9	-1,9
34	32	Conner Peripherals	2200	2240	-1,8
35	38	CA	2054,8	1770,8	16,0
36	35	Tandem	2023	2066,3	-2,1
37	40	Quantum	2002	1536,2	30,3
38	52	AST Research	1971	1140,5	72,8
39	31	Cap Gemini	1946,9	2252,9	-13,6
40	48	Acer	1898	1192,2	59,2

1993	1992	Firma	Przychody w 1993 r.	Przychody w 1992 r.	zmiana [%]
41	47	Intel	1844,2	1230	49,9
42	39	Seiko Epson	1768,6	1649,8	7,2
43	54	Gateway 2000	1731,7	1107,1	56,4
44	46	Oracle	1692,9	1310	29,2
45	27	Amdahl	1680,5	2524,7	-33,4
46	49	First Data	1500	1205	24,5
47	53	Samsung	1494,4	1117,7	33,7
48	41	StorageTek	1405	1521,5	-7,7
49	45	Lexmark	1350	1350	0
50	59	Western Digital	127	943,4	34,4

Widzimy, że IBM utrzymała swą pierwszą pozycję. Mimo spadku przychodów o 2,8% są one prawie trzykrotnie wyższe od następnej na liście Fujitsu. Tak więc kolejność pierwszej trójki nie zmieniła się i po IBM idą dwie firmy japońskie, które zwiększyły w 1993 roku swoje przychody o ponad 8%. Pierwsza zmiana nastąpiła na 4-tej pozycji, gdzie dynamicznie rozwijający się Hewlett-Packard (wzrost przychodów o 23%) wyprzedził nie mogącego się wciąż przystosować do nowych warunków Digitala (spadek przychodów o 3,7%). Dalsze dwie pozycje w tabeli nie uległy zmianie, przy czym Hitachi wykazała najwyższy wzrost przychodów wśród dotychczas wymienionych firm japońskich (o 11,2%), a AT&T - największy dotychczas notowany spadek (o 5,5%). Na ósmym miejscu znalazła się Toshiba awansując w tabeli o dwa miejsca dzięki wzrostowi przychodów o 18,4%. Na 9 miejscu znalazła się amerykańska firma EDS (Electronic Data System Corp. z Plano w Teksasie) awansując aż o 6 miejsc w tabeli przy wzroście przychodów zaledwie o 3,5%, podczas gdy Apple przesunęła się w górę tylko o jedno miejsce przy wzroście dochodów o ponad 10%. Spadły natomiast firmy zachodnioeuropejskie jak Siemens, Olivetti, Bull, ICL i Cap Gemini o 2-8 miejsc w tabeli przy spadku przychodów o 10-14%, oraz niektóre japońskie (Matsushita i Oki), przy mniejszym spadku. Z firm amerykańskich znaczniejsze tendencje spadkowe wykazały Unisys (o trzy miejsca i 8,1%), Amdahl (18 miejsc i 33,4%), a z drugiej pięćdziesiątki Maxtor (8 miejsc i 11,4%) i Intergraph (6 miejsc i 10,8%).

Uwzględniając pełną, 100-pozycyjną listę, tabela 3 przedstawia 10 firm o najwyższym procentowym wzroście przychodów z techniki informatycznej. Znalazły się tu przede wszystkim małe firmy amerykańskie, z których trzy pierwsze po raz pierwszy zjawily się w 1993 roku na tej liście, a jedynie Compaq znajduje się w pierwszej piętnastce. Wracając do tabeli 2, to pojawiło się w niej 8 nowych firm - wspomniane 3 z tabeli 3 (odpowiednio na pozycjach 63, 75 i 81) oraz 5 firm na pozycjach poniżej 93. Firma Science Application przesunęła się o 27 pozycji w górę tabeli, Martin Marietta o 25 pozycji, 3 Com o 15 i AST Research o 14. Wśród firm, które pozostały w pierwszej setce, ale znacznie obniżyły swe przychody w stosunku do 1992 roku, należy wymienić Wang (spadek o 39,3% i 23 pozycje) i wspomnianą Amdahl oraz Memorex Telex (odpowiednio 24% i 18). Ponadto nastąpiły niewielkie zmiany, a 5 firm w pierwszej siódemce nie zmieniło swych pozycji.



Tabela 3

Lp.	Firma	wzrost wpływów [%]
1	EMC	124,2
2	Cisco	96,1
3	SynOptics	81,2
4	Compaq	75,6
5	AST Research	72,8
6	Science Application	63,2
7	Acer	59,2
8	Martin Marietta	58,5
9	Gatewav 2000	56,4
10	Intel	49,9

Tabela 4 podaje 30 czołowych firm północnoamerykańskich. Znajduje się w niej nieco więcej danych niż w tabeli 2, głównie jaki procent przychodów firmy stanowi technika informatyczna, a także dane o zysku w latach 1992-93.

Tabela 4

Miejsce 93	92	Firma	Przychody		Δprz. [%]	info. [%]	Zysk		Δzysk [%]
			1993	1992			1993	1992	
1	1	IBM	62716	64520	-2,8	100	-8101	-6870	-17,9
2	3	HP	15600	12688	23,0	76,8	1200	549	118,6
3	2	Digital	13637	14162	-3,7	100	-72	-2501	97,1
4	4	AT&T	9860	10450	-5,6	14,7	-3974	3807	-204,4
5	7	EDS	8507,3	4870	74,7	99,4	724	635,5	13,9
6	6	Apple	7900	7173,3	10,1	100	86	525,7	-83,6
7	5	Unisys	7200,5	7832	-8,1	93	565,4	361	56,6
8	8	Compaq	7200	4100	75,6	100	462	213,2	116,7
9	9	Sun	4493	3832	17,2	100	188,2	127,9	47,2
10	10	Microsoft	4110	3253	26,3	100	1036	833	24,4
11	12	Xerox	3330	3016	10,4	22,8	620	-1020	160,8
12	11	Scagate	3114	3079,4	1,1	100	89,1	75,7	17,7
13	13	Anderson	2876,3	2720	5,7	100	-	-	-
14	20	Dell	2870	2013,9	42,5	50	-35,8	101,6	-135,2
15	15	CSC	2502	2474,4	1,1	100	98,1	75,7	17,7
16	18	A.P.	2339,1	2075	12,7	100	310,8	272	14,3
17	21	TRW	2251	1855	21,3	28,3	195	194	0,5
18	16	International Telecom	2250	2100	7,1	27,6	-878	536	-263,8
19	17	Conner Periph	2200	2240	-1,8	100	-445	121	-467,8
20	22	CA	2054,8	1770,8	16,0	100	341,6	204,2	67,3

Miejsce 93 92	Firma	Przychody		$\Delta$ pr. [%]	infor. [%]	Zysk		$\Delta$ zysk [%]
		1993	1992			1993	1992	
21 19	Tandem	2023	2066,3	-2,1	100	-510	70,9	-818,9
22 23	Quantum	2002	1536,2	30,3	100	-9,5	84,8	-111,2
23 32	AST Research	1971	1140,5	72,8	100	-50	57,6	-186,8
24 29	Intel	1844,2	1230	49,9	21	2295	1066,5	115,2
25 33	Gateway 2000	1731,7	1107,1	56,4	100	100,1	70	43,0
26 28	Oracle	1692,9	1310	29,2	100	197,8	89	122,2
27 14	Amdahl	1680,5	1310	29,2	100	197,8	89	122,2
28 30	First Data	1500	1205	24,5	100	173	141	22,7
29 24	Storage Tech	1405	1521,5	-7,7	100	-78	15,5	-603,2
30 27	Lexmark	1350	1350	0	75	-	-	-

Te dodatkowe dane pokazują, że 39% spośród wymienionych czołowych firm poniosło w 1993 roku straty (w porównaniu do 14% w 1992 roku), a 43% wykazało pogorszenie zysku lub zwiększenie strat. Nawet firmy zwiększające znacznie swe przychody i awansujące na liście, jak Dell czy AST Research, poniosły w 1993 roku straty. W wartościach bezwzględnych największe straty poniosła IBM (ponad 8 mld dolarów), choć w ostatnich kwartałach i tu wystąpiła poprawa. Procentowo największą nickorzystną zmianę odnotował Amdahl (ponad 82 razy). Największy zysk bezwzględny osiągnął Intel (prawie 2,3 mld dolarów), a procentowo - Xerox (ponad 160%). Według pełnej listy amerykańskiej Cray uzyskał ponad 5-krotne zwiększenie zysku, a EMC - ponad 3-krotne.

W cytowanej części tabeli 70% firm to firmy czysto informatyczne. Najmniejszy udział tej techniki widzimy u AT&T (14,7%); w pełnej tabeli tylko 1,6% notowane jest u Bociinga. Zmiany w tej tabeli podobne są do zmian w Tabeli 2 uwzględniając wycielimowanie firm z innych kontynentów.

Przejdźmy teraz do poszczególnych dziedzin rozwoju informatyki. Nie będziemy tu przytaczać tabel dla wszystkich dziesięciu wybranych dziedzin. Natomiast sumując przychody pierwszych dziesięciu firm w danej dziedzinie uzyskamy względną ocenę wagi danej dziedziny w 1993 roku. Przedstawimy także procentowe zmiany tych przychodów w stosunku do roku poprzedniego. Pokazuje to tabela 5. Sumy przychodów wyrażone są w mln USD.

Jak widać, największy procentowy przyrost wykazały komputery osobiste (36,3%), a następnie komunikacja (17,2%) i usługi (13,9%). Zmniejszenie natomiast wykazały duże systemy (37,6%), oraz w niewielkim stopniu stacje robocze (3,6%) i urządzenia peryferyjne (2,5%). Przyczyną spadku sprzedaży dużych systemów jest odwrócenie proporcji pomiędzy zapotrzebowaniem na moc obliczeniową i stosunkiem wydajności do ceny. W rezultacie IBM zmniejszyła sprzedaż dużych maszyn o 47%, lecz nadal jej udział w tej dziedzinie wynosi 5,5%. Podobnie (o 42%) wystąpił spadek u Amdahla i w mniejszym stopniu (o 16%) w firmie Unisys. Rozwój w dziedzinie dużych systemów odnotował Intel (54%) oraz Cray, PP i Digital (odpowiednio 18, 11 i 10%).



Obserwuje się tu zastępowanie dużych maszyn komputerami unixowymi. Cena za milion operacji na sundę spadła w tej dziedzinie z 45 do 30 tys. dolarów, a niektóre przewidywania mówią o 4000 dolarów w 1996 roku.

Tabela 5

L.p.	Dziedzina	Przychody 10 gl. firm	Zmiana [%]
1	Usługi	33750	13,9
2	Komputery osobiste	33400	36,3
3	Urządzenia peryferyjne	29430	-2,5
4	Oprogramowanie	22940	7,3
5	Utrzymywanie systemów	20290	1,3
6	Średnie systemy	12960	6,3
7	Komunikacja	11860	17,2
8	Stacje robocze	9040	-3,6
9	Duże maszyny	7900	-37,6

Taki rozwój sytuacji pociągnął za sobą redukcję zatrudnienia w ośrodkach wytwarzających duże systemy i zwrot ku tańszym technologiom. Pierwsze wyniki za rok 1994 mówią, że dało to rezultaty i dalszy spadek w tej dziedzinie został zahamowany.

W dziedzinie średnich systemów największe sukcesy odnotował HP (58% wzrost przychodów i osiągnięcie 16% udziału rynku). Wyższy procent wzrostu sprzedaży odnotowała tylko Data General, lecz jej udział wynosi tylko 4%. IBM zachowała pierwszą pozycję z 43% udziałem, ale jej przychody w tej dziedzinie nie zmieniły się od poprzedniego roku. Spadek natomiast wykazał Digital (o 20% przy 15% udziale) i Stratus (odpowiednio 19% i 3%). Największy rozwój wykazały tu systemy unixowe często wykorzystywane do zarządzania bazami danych.

W zakresie oprogramowania tylko IBM wykazała niewielki (4%) spadek przychodów, pozostając jednak na pierwszym miejscu z 36% udziałem. Najlepiej wypadł Microsoft (26% wzrost sprzedaży i 12% udziału) oraz Oracle (odpowiednio 29% i 4%) i WordPerfect (22% i 2%). Interesujące, że Borland znalazł się poza pierwszą dziesiątką. Wzrósł tu udział systemów klient-serwer, a także pojawiły się pierwsze systemy ukierunkowane na obiekty, których udział będzie rósł w następnych latach. Obserwuje się tu nadal koncentrację dużych firm jak zakupienie przez Novellę WordPerfect i Quattro Pro.

W dziedzinie o największych dochodach - usługach - nie odnotowano w 1993 roku spadku przychodów w pierwszej dziesiątce firm. Największy przyrost wykazała IBM (32% przy 21% udziale), potem First Data (24% i 3%) oraz Digital i Unisys (po 19% przy udziałach odpowiednio 4 i 3%). Posiadająca 18% udziału EDS wykazała 4% wzrost, a najmniejszy (1% przy 5% udziale) przyrost przychodów miała firma CSC. Tu również widać rozwój techniki klient-serwer i firmy, które potrafiły zaoferować tańsze rozwiązania, uzyskały większy wzrost.

Komunikacja zajęła drugie miejsce wśród najszybciej rozwijających się dziedzin, choć mająca największy udział w rynku (23%) IBM nie odnotowała wzrostu przychodów. Prawie dwukrotny wzrost (96%) wykazała firma Cisco z 7% udziałem, a następnie SynOptics (81% i 5% udziału) i Cabletron (49% i 4%). Duże firmy

telekomunikacyjne jak AT&T oraz Northern Telecom mające po 16-19% udziału w rynku wykazały wzrost 5-7%. Można tu mówić o pierwszych transakcjach dotyczących tzw. autostrad informatycznych, lecz większy rozwój przewiduje się tu w następnych latach.

Dziedzina stacji roboczych zajęła przedostatnie miejsce (przed wielkimi maszynami) zarówno, jeśli chodzi o wartość przychodów jak i współczynnik rozwoju (spadek o 3,6% w stosunku do roku 1992). Tylko HP, zajmujący z 22% udziałem drugie miejsce w tej branży, uzyskał wzrost przychodów o 55%. Wzrost o 28% wykazała charakteryzująca się 9% udziałem firma Silicon Graphics, a o 17% lider stacji roboczych - Sun, posiadający 26% udział. Pozostałe firmy pierwszej dziesiątki miały bądź nieznaczny wzrost (Wyse o 4%), bądź spadek przychodów. Rekordzistką okazała się firma Intergraph, w której spadek wyniósł 66% (udział 2%), a po niej Computervision (odpowiednio 40% i 1%) i Unisys (31% i 4%). Również IBM mająca 13% udział rynku odnotowała 25% spadek. Tańsze stacje robocze wypierane są przez bardziej rozbudowane systemy komputerów osobistych, które stają się konkurencyjne pod względem ceny i parametrów. Rozwijają się natomiast stacje bardziej złożone, dla których uzyskano odpowiednio wysoki stosunek wydajności do ceny.

Jeśli chodzi o urządzenia peryferyjne, to ta trzecia pod względem przychodów dziedzina wykazała w sumie niewielki spadek w stosunku do poprzedniego roku, a to z przyczyny spadku przychodów jej lidera - IBM, która mając 5% udział odnotowała 27% spadek w 1993 roku. Podobnie było u Digitala (spadek 22 przy 6% udziale) oraz niewielki spadek (2% przy 6% udziale) w firmie Conner Peripherals. Pozostałe firmy pierwszej dziesiątki odnotowały wzrost przychodów - największy firmy Lexmark i Quantum (odpowiednio 34 i 30% przy 5% udziale każdej z firm), a HP i Xerox po 13% (odpowiednio przy udziale 13 i 8%). Szybko rozwijającymi się obszarami są tu usprawnione systemy dyskowe, w tym dyski optyczne stające się powoli standardem w komputerach osobistych.

Utrzymywanie systemów znalazło się pośrodku wyodrębnionych dziedzin z niewielkim (1,3%) współczynnikiem wzrostu przychodów. Liderem jest tu również IBM, która przy udziale 33% wykazała 4% spadek. Większy spadek wykazały firmy Unisys i Memory Telex (odpowiednio 14 i 10% przy 2-3% udziale). Natomiast firma Sun wykazała 15% wzrost przy 2% udziale, a HP 5% wzrost przy 13% udziale. Przewiduje się, że dochody z tej dziedziny będą maleć w najbliższych latach, gdyż nowe systemy wymagają mniejszych nakładów na ich utrzymanie. Występuje też zjawisko obsługi systemów jednych firm przez inne. Natomiast część usług oferowanych dotychczas za darmo została zaliczona do płatnych.

Jak zapowiedziano w tytule szczegółowej zajmijmy się dziedziną komputerów osobistych. Dysponujemy tu wcześniejszymi materiałami dotyczącymi roku 1992 i pierwszej połowy 1993 [2]. W roku 1992 niektórym firmom udało się podwoić wartość sprzedaży, a wśród firm, które pozostawały w pierwszej pięćdziesiątce producentów komputerów osobistych zarówno w 1991 jak i 1992 roku, przeważały te, które poprawiły swą pozycję.

W tabeli 6 [2] przytaczamy listę pierwszych pięćdziesięciu wytwórców w 1992 roku według wartości sprzedanych systemów osobistych (w milionach dolarów USA), podając jednocześnie ich pozycje w analogicznym rankingu z 1991 roku, wartość sprzedanych wówczas systemów, zmianę procentową oraz sumy uzyskane ze sprzedaży



w pierwszej połowie 1993 roku. Dla porównania podano też ilości systemów (w tys. szt.) sprzedanych w 1992 i 1991 roku oraz odpowiadające temu zmiany procentowe.

Pierwszych pięćdziesięciu producentów komputerów osobistych.

**Tabela 6**

Kolejność 92 91		Firma	Sprzedaż [mln USD]		Zmiana [%]	Sprzedaż I poł. 93	Ilość [tys. szt.]		Zmiana [%]
			1992	1991			1992	1993	
1	1	IBM	7655	8505	-10,0	2915	3306	3058	8,1
2	3	Apple	5412	4900	10,5	3836	2910	2261	28,7
3	4	Compaq	4100	3271	25,3	3244	1560	998	56,3
4	2	NEC	3987	5280	-24,5	2221	1406	1438	-2,2
5	5	Fujitsu	2618	2342	11,8	1414	230	190	21,2
6	6	Toshiba	1949	2045	-4,7	974	509	566	-10,1
7	18	Dell	1813	667	171,6	1373	774	267	190,0
8	7	Olivetti	1349	1578	-14,5	527	631	563	12,1
9	13	AST Research	1141	827	37,9	780	437	251	55,7
10	25	Geteway 2000	1107	627	76,6	786	437	251	74,3
11	9	AT&T/NCR	999	950	5,2	493	145	216	-32,7
12	12	Hitachi	892	825	8,1	497	19	29	-34,5
13	16	Acer	890	709	25,5	502	588	476	23,5
14	19	Packard Bell	879	641	37,1	514	700	515	36,0
15	17	Seiko Epson	741	708	4,8	400	213	292	-27,2
16	20	Canon	741	613	21,0	488	6	13	-53,9
17	8	Commodore	732	1039	-30,0	313	1195	1194	0,1
18	15	Hewlett-Packard	725	715	1,4	310	302	177	71,1
19	33	ICL	713	297	139,8	604	202	-	-
20	-	Vobis	686	-	-	-	396	-	-
21	11	Groupe Bull	686	890	-22,9	266	495	468	5,7
22	14	Tandy	650	750	-13,3	284	571	477	19,7
23	21	Siemens Nixdorf	526	362	45,4	339	229	178	28,7
24	23	CompuAdd	524	513	2,2	-	206	141	46,0
25	39	Mitac	522	441	18,4	274	204	194	5,3
26	26	Digital	450	215	109,3	450	203	61	232,7
27	10	Unisys	434	900	-51,8	186	152	96	58,7
28	50	Northgate	400	167	139,3	-	-	-	-
29	36	Hyundai Electron.	391	269	45,6	213	235	304	-22,8
30	47	Tatung	366	282	29,5	151	660	625	5,6
31	31	Everex	350	307	13,9	-	-	201	-
32	27	Ricoh	340	409	-16,9	204	35	39	-9,1
33	38	Mitsubishi	311	242	28,2	-	-	-	-
34	45	First Internat.	301	204	47,4	184	-	-	-

Kolejność 92 91		Firma	Sprzedaż [mln USD]		Zmiana [%]	Sprzedaż I poł. 93	Ilość [tys. szt.]		Zmiana [%]
			1992	1991			1992	1993	
35	41	Trigem	286	217	31,8	-	-	-	-
36	24	Tandon	285	461	-38,3	-	132	190	-30,4
37	48	Samsung	267	199	34,1	-	-	-	-
38	43	Memorex Telex	257	255	0,7	97	120	70	70,7
39	-	Compudync	238	-	-	-	-	-	-
40	-	Copam	225	-	-	-	-	150	-
41	28	Wang	220	340	-35,3	-	104	107	-3,0
42	46	Matsushita	215	203	6,1	-	-	-	-
43	40	Zeos	206	231	-10,7	113	-	-	-
44	35	Amstrad	202	272	-25,5	87	220	254	-13,3
45	37	ALR	200	227	-12,1	80	120	142	-14,9
46	42	Tulip Computers	189	215	-12,1	101	97	112	-13,6
47	-	Goldstar	182	194	-6,4	88	-	-	-
48	-	Leading Edge	170	90	88,9	85	125	72	73,6
49	-	Texas Instruments	160	150	6,7	121	65	47	38,1
50	-	Xerox	144	-	-	-	-	-	-
Razem			48824						

Wskaźniki rozwoju, jakie osiągnęły czołowe firmy, są lepsze niż w ubiegłych latach, a suma uzyskana przez wszystkie 50 firm (48,8 mld dolarów) też jest nowym rekordem. Jednak łączny wzrost sprzedaży 50 najlepszych firm wyniósł tylko 5,6%, co jest rezultatem ostrej walki cenowej ograniczającej zyski. Na przykład firma Apple, która wykazała 10% wzrost wpływów w 1992 roku, miała 44% marżę zysku w pierwszym kwartale tego roku. Natomiast w 1993 roku, mimo wzrostu sprzedaży o 21%, zysk zmniejszył się do 25%. Jeszcze dobitniej widać to na przykładzie firmy Dell, która w 1992 roku osiągnęła wpływy ponad 1,8 mld dolarów - wzrost o ponad 170% w stosunku do roku 1991. Jeśli chodzi o ilość sprzedanych systemów wzrost ten był jeszcze większy i wyniósł 190%. Jednakże marża zysku spadła z 26% w II kwartale 1992 roku do zaledwie 6,5% w IV kwartale. Taka sytuacja spowodowała kłopoty takich firm jak Everex i Tandon.

Ceny komputerów osobistych spadły gwałtownie. Jeszcze niedawno sięgały one 10 tys. dolarów, a obecnie systemy oparte na procesorze 486 takie jak ValuePoint IBM, Dimension Della lub ProLinea firmy Compaq, sprzedawane są poniżej 2500 dolarów. Firmy takie jak Dell i Gateway 2000 znalazły się po raz pierwszy w pierwszej dziesiątce eliminując z niej AT&T, Commodore, International i Unisys Corp.

Pozostająca w 1992 roku na pierwszej pozycji IBM traci dystans do swoich rywali i w pierwszej połowie 1993 roku została wyprzedzona przez Apple i Compaq. Próbuje ona ratować sytuację tworząc podległe sobie organizacje sprzedaży komputerów osobistych jak IBM PC Direct i Ambra, które przejęły znaczną część tej roli.



Tabela 7 pokazuje 20 firm o największych wskaźnikach wzrostu sprzedaży w 1992 roku w stosunku do roku 1991 wraz z procentową wartością tych wskaźników.

Firmy o najwyższym przyroście sprzedaży 1992/1991. Tabela 7

L.p.	Nazwa firmy	Wskaźnik wzrostu [%]
1	Dell	171,6
2	ICL	139,8
3	Northgate	139,3
4	Digital	109,3
5	Leading Edge	88,9
6	Gatewa 2000	76,6
7	First International	47,4
8	Hyundai Electronics	45,6
9	Siemens Nixdorf	45,4
10	AST Research	37,9
11	Packard Bell	37,1
12	Samsung	34,1
13	Trigem	31,8
14	Tatung	29,5
15	Mitsubishi	28,2
16	Acer	25,5
17	Compaq	25,3
18	Canon	21,0
19	Mitac	18,4
20	Everex	13,9

Wśród występujących tu firm warto zwrócić uwagę na Digital, który ostatnio podejmuje energiczne wysiłki w zakresie sprzedaży komputerów osobistych. Wiadomo też, że podobną akcję prowadzi Hewlett-Packard.

Listę 10 głównych producentów komputerów osobistych w 1993 roku podaje tabela 8 [1].

Cała pierwsza dziesiątka firm odnotowała wzrost przychodów. Najwięcej (76% przy 21% udziale) uzyskał Compaq, który znalazł się na drugim miejscu tabeli, a następnie AST Research (odpowiednio 72% i 6%) oraz Gateway 2000 (56% i 5%). Czwartą firmą w kolejności stopnia wzrostu przychodów jest Dell (43% i 8%), lecz jest to prawie trzykrotnie mniej niż przyrost w roku 1992 wynoszący 126%. Utrzymująca nadal pierwszą pozycję IBM wykazała wzrost o 23% przy 29% udziale, natomiast najniższy poziom wzrostu wykazały IT&T (2% i 3%) oraz Apple (9% i 17%). Wśród najnowszych wyrobów w tym zakresie widzimy wiele komputerów typu notebook jak Compaq Concerto, IBM ThinkPad i AST Research Bravo.

Jak słychać z ostatnich doniesień [3] Compaq w drugim kwartale 1994 roku dostarczył najwięcej komputerów sprzedanych w USA (14,3%), na drugim miejscu jest Apple (11,2%), potem Packard Bell Electronics (9,3%), a dopiero potem IBM (7,9%).

Główni dostawcy komputerów osobistych w 1993 roku.

Tabela 8

Miejsce	Firma	Przych.93 [mln USD]	Przych.92 [mln USD]	Zmiana [%]	Udział w [%]
1	IBM	9728	7887	23	29
2	Compaq	7200	4100	76	21
3	Apple	5925	5412	9	17
4	Dell	2583	1812,5	43	8
5	AST Research	1966,9	1140,5	72	6
6	Gateway 2000	1731,7	1107,1	56	5
7	Packard Bell	1250	878,8	42	4
8	AT&T	1017	998,9	2	3
9	Hewlett-Packard	889,2	725	23	3
10	Digital	609	450	35	2

Jeśli chodzi o stosowane w komputerach osobistych mikroprocesory, to dominuje tu rodzina X86 Intelu, która stanowiła w 1993 roku 87% wszystkich sprzedanych mikroprocesorów [4]. Najwięcej było procesorów 486DX (prawie 37% wszystkich X86), potem 486SX (niespełna 33%) i 386 (prawie 30%). Procesorów Pentium sprzedano w 1993 roku tylko 210 tysięcy, co stanowiło zaledwie 0,5% X86. Według przewidywań firmy Mercury Research w roku 1994 proporcje te ulegną zmianie i przy ogólnym wzroście sprzedanych mikroprocesorów o 18,9% do 53,1 mln, wzrost procesorów X86 będzie jeszcze wyższy (21,4%), a wśród nich największy wzrost (ponad 10-krotny) wykaże Pentium, którego sprzedaż wyniesie 2,13 mln sztuk (4,6% rodziny X86), ale najliczniejszym typem będzie nadal procesor 486DX (19,5 mln - 41,3%), mimo wolniejszego tempa wzrostu sprzedaży (poniżej 36%). Według tych samych źródeł 1995 rok ma przynieść dalsze zmiany. Wzrost sprzedaży w stosunku do 1994 roku ma być niższy (10,7% ogólnie i 6,5% dla X86). W większym zakresie wykorzystywać się będzie mikroprocesory RISC, takie jak PowerPC triumwiratu Apple/IBM/Motorola, Alpha Digitala i układu Mips firmy Silicon Graphics. Jednakże procesory X86 stanowiąc będą nadal 88,6% wszystkich mikroprocesorów, a więc więcej niż w 1993 roku i tylko o 0,3% mniej niż w roku 1994. Tak samo największy wzrost wykaże Pentium, ale tym razem już tylko niespełna trzykrotny w stosunku do poprzedniego roku (5,85 mln co stanowi 11,6% X86). Sprzedaż procesora 486DX ma wzrosnąć o 46,4% do 28,6 mln i stanowić on ma aż 56,8% wszystkich X86. Natomiast spadać już będzie sprzedaż zarówno 486SX (o 31% do 13,2 mln), jak i 386 (o 58,3% do 2,7 mln). Stanowiąc one wówczas mają tylko, odpowiednio, 26,3% i 5,3% wszystkich procesorów X86.

Zę względu na dominującą rolę procesorów X86, w Tabeli 9 zestawiono główne dane o tych układach [5].

Pierwsze procesory Pentium (568) pracowały z częstotliwością zegara 60 i 66 MHz. W marcu 1994 roku [6] wprowadzono nowe wersje tego mikroprocesora o częstotliwości zegara 90 i 100 MHz. Pierwszy kosztował wówczas 849 dolarów, a drugi 995. Jednocześnie zaoferowano procesor 486DX4 o częstotliwości zegara 75 i 100 MHz w cenie odpowiednio 475 i 580 dolarów. O dalszej obniżce cen procesorów Intelu doniesiono w trzecim kwartale 1994 roku [7].



## Sprzedaż procesorów firmy Intel.

Tablica 9

Symbol	Data początku sprzedaży	Cena początkowa [ USD ]	Cena w V.1993 [ USD ]	Wydajność początkowa [mln. op./s]	Wydajność max.	Ilość tranzystorów
8086	VII.1978	360	-	0,33	0,75	29000
286	II.1982	360	8	1,2	2,66	134000
386	X.1985	299	91	5	11,4	275000
486	VIII.1989	950	317	20	54	1200000
586	III.1993	900	900	112	112	3100000

Konkurencję dla Pentium stanowi wspomniany już mikroprocesor PowerPC 601 [6], który zajmuje mniejszą powierzchnię (121 mm<sup>2</sup> w porównaniu z 264 mm<sup>2</sup> dla Pentium) i kosztuje tylko 450 dolarów.

Wracając do omawianych za [1] dziewięciu dziedzin rynku informatycznego i uwzględniając pierwsze trzy firmy we wszystkich wymienionych dziedzinach, IBM znajduje się w siedmiu z nich na pierwszym miejscu, a w dwu pozostałych na trzecim. HP pojawia się tu czterokrotnie - trzy razy na drugim miejscu i raz na trzecim, a Digital dwukrotnie - raz na drugim i raz na trzecim. Pozostałe 12 firm występują tylko raz.

Podsumowując światowy rynek można stwierdzić, że rok 1993 był kolejnym trudnym rokiem w rozwoju przemysłu informatycznego na świecie, który w sumie przyniósł współczynniki rozwoju lepsze niż w innych dziedzinach gospodarki, ale znacznie niższe aniżeli posiadał je przed kilkoma laty, w okresie swego bujnego rozwoju.

Dla poszczególnych firm sytuacja przedstawiała się różnie. Można mówić o pierwszych oznakach zahamowania regresu dużych firm i bardziej dynamicznym rozwoju tych, które szybciej dostosowały się do istniejących trendów.

## Największe firmy informatyczne w Polsce.

Tabela 10

Miejsce 93	Miejsce 92	Firma	Siedziba	Obroty [mld zł]		Δobr. [%]	Ilość prac.	
				1992	1992		1993	1992
1	2	Optimus (holding)	Nowy Sącz	1912,1	648	195	450	86
2	3	JTT (holding)	Wrocław	1895,7	610	211	210	47
3	1	Softronic Logistics	Wrocław	976	890	9,7	58	101
4	4	IBM Polska	Warszawa	932,34	495	88	100	100
5	7	HP Polska	Warszawa	816,55	346,4	136	85	70
6	6	ICL	Warszawa	530	351	51	230	150
7	8	Hector	Warszawa	530	236,4	124	37	29
8	15	Bull Polska	Warszawa	447,9	162	177	24	22
9	5	Protech	Warszawa	430	400	7,5	180	180
10	24	DHI/Computer 2000	Warszawa	414,33	92	350	47	28
11	11	InterAms II	Warszawa	390,69	206,8	89	177	128
12	10	ABC Data	Warszawa	382,11	223	71	63	50

Miejsce 93 92		Firma	Siedziba	Obroty [mld zł] 1992 1992		Δobr. [%]	Ilość prac. 1993 1992	
13	18	ComputerLand Poland	Warszawa	366,57	128	186	90	60
14	12	Prokom	Gdynia	360,91	190	90	149	78
15	28	Everest	Gdynia	349,27	80	327	150	60
16	72	Incom	Wrocław	311,82	27	1055	36	42
17	16	NTT System	Warszawa	293,56	161	82	38	41
18	13	Techmex	Bielsko-Biała	273	170	61	85	100
19	19	Baza	Warszawa	255,21	120	113	56	45
20	9	Unilot	Warszawa	241,33	229,5	5,2	81	60
21	-	Escom Computer Polska	Swarzędz	231,75	-	-	88	-
22	64	Dell Polska	Warszawa	215,5	32,3	565	29	14
23	17	COIG	Katowice	205,43	150	37	1055	1200
24	34	Digital Equip. Pol.	Warszawa	199,5	70	185	70	70
25	-	California Computer	Warszawa	191,59	-	-	16	-
26	26	System 3000	Kraków	190	91,8	107	45	21
27	37	Tex	Poznań	166,84	61,2	173	22	9
28	-	Business Power Sys.	Warszawa	161,32	-	-	85	-
29	-	Datex	Łódź	159,12	-	-	38	-
30	-	Vobis	Szczecin	158	-	-	45	-
31	22	Ken	Wrocław	154,84	101	53	68	75
32	32	Inwar	Sieradz	152,26	70	118	104	95
33	42	MSP	Warszawa	150,64	53,2	184	38	25
34	35	Sad-Apple IMC Pol.	Warszawa	141,41	67,5	109	42	35
35	-	Techmex Internat.	Katowice	140	-	-	35	-
36	30	Qumak	Kraków	138,83	77,8	78	67	60
37	21	Comptrade	Warszawa	124,43	102,6	21	34	27
38	22	Hyundai Seiko	Warszawa	123	96	28	15	15
39	54	FF Computers	Bielsko-Biała	117,3	40,1	193	18	10
40	41	River	Kraków	116,3	53,3	118	56	51
41	27	Apexim	Warszawa	114,53	91,7	25	121	40
42	-	Ab	Wrocław	109,12	-	-	14	-
43	33	CSBI	Warszawa	106,16	70	52	127	86
44	58	Koma	Katowice	105,45	36	193	47	30
45	-	Bajtex	Zabrze	102,8	-	-	24	-
46	-	Consoft	Kraków	102,29	-	-	16	-
47	-	Centrum Inf. En.	Warszawa	101,56			-	322
48	40	Unox	Brzeg Dln.	100,82	54	87	41	35
49	-	Olivetti	Warszawa	100	-	-	8	-
50	43	ATM	Warszawa	97,46	51,3	90	110	99



Pozostaje nam jeszcze krótkie przedstawienie polskiego rynku informatycznego. Wykorzystamy tu częściowo dane zawarte w raporcie [8].

Porównując współczynniki rozwoju firm światowych i polskich możemy stwierdzić, że mamy tu do czynienia z szybszym rozwojem (co częściowo spowodowane jest większą inflacją) oraz większą ilością nowych firm pojawiających się na rynku. Uwzględniając wszystkie 200 firm z raportu [8], tabela 11 zestawia najszybciej rozwijające się firmy.

Tabela 11

L.p.	Firma	wzrost obrotów [%]	Awans w rankingu
1.	Incom	1055	56
2.	Dell Polska	565	42
3.	DHI/Computer 2000	350	1
4.	Everest	327	13
5.	Surfland	270	22
6.	JTT	211	1
7.	Solidex	210	13
8.	Optimus	195	1
9.	FF Computers	193	15
10.	Koma	193	14
11.	Computerland Poland	186	5
12.	Digital Equipment Polska	185	10
13.	MPS	184	9

Rzadziej mamy do czynienia z firmami, które zanotowały spadek obrotów - jest ich tylko siedem na 98, dla których podano dane za rok 1992 i 1993. Największy spadek (o 55%) odnotowała firma Comtech-Wektor i spadła w rankingu o 55 pozycji, następnie Bonair (29% i 44 pozycje), Siemens-Nixdorf (12% i 31), NCR (11% i 37), Rex (9% i 43), Intern. Comp. Service (5% i 57) i Efekt (1,2% i 47). Natomiast sam spadek pozycji na liście rankingowej jest częściej występującym zjawiskiem niż poprawa pozycji. Na 94 firmy, których pozycje podano w 1992 i 1993 roku, 23 odnotowało poprawę pozycji, sześć zachowało ją bez zmian, a 65 wykazało spadek. Najwięcej awansów pozycji zanotowano w pierwszej pięćdziesiątce firm (15), mniej (7) w drugiej, tylko jeden w trzeciej i żadnego w czwartej. Natomiast spadków najwięcej było w drugiej pięćdziesiątce (27), po 18 w pierwszej i trzeciej, oraz dwa w czwartej. Należy jednak zwrócić uwagę, że jak już wspomniano, ponad połowa pierwszych polskich 200 firm informatycznych to firmy nowe, które nie istniały w 1992 roku. Pierwsza z nich pojawia się już na 21 pozycji (Escom Computer Poland), od razu z blisko 200 mld zł obrotów. W pierwszej pięćdziesiątce jest ich jedenaście, w drugiej - 16, w trzeciej - 31, a w czwartej aż 148. Można stąd wyciągnąć wniosek, że polski rynek informatyczny jest rynkiem jeszcze nowym, na którym dominują małe firmy.

**Literatura:**

- [1] Users to Vendors: Quality, Quality, Quality; Datamation June 15, 1994, str. 38 - 64
- [2] Mice Riccuti, The PC 50, Datamation, December 1, 1993, str. 36-8
- [3] IBM - Compaq, Rzeczpospolita, Nr 197/94, str. 7
- [4] Market Forecast 1994, Electronics, 10 January 1994, str. 5-14
- [5] Tom R. Halfhill, Intel Launches Rocket in a Socket, Byte, May 1993, str. 92
- [6] Computerworld Nr 16/140, str. 1
- [7] PC Courier Nr 19/94, str. 44
- [8] Raport specjalny Computerworld: Top 200 - Polski rynek komputerowy.





*Jeśli:*

- *stosujecie egzaminy testowe,*
- *przeprowadzacie egzaminy ustne,*
- *organizujecie szkolenia*

*wykorzystajcie oferowane przez nasz Instytut oprogramowanie narzędziowe*

## **Jądro Komputerowego Systemu Oceny Poziomu Wiedzy**

Oprogramowanie to służy do tworzenia i obsługi programów komputerowych będących testami lub programami edukacyjnymi, a przeznaczone jest dla komputerów klasy IBM PC i działa pod systemem DOS ver. 5.0. JKSOPW pozwala na realizowanie nauczania i egzaminowania bez znajomości języków programowania.

\* Testy lub programy nauczające mają formę zorganizowanego dialogu z użytkownikiem komputera, gdzie powtarzają się:

- przedstawienie problemu-pytania,
- wybór przez użytkownika jednej z przedstawionych odpowiedzi,
- ewentualne skomentowanie tego wyboru i przejście do następnego problemu-pytania.

Pytania, odpowiedzi i komentarze mogą zawierać ilustracje.

\* Każdy problem-pytanie ma określoną wagę (stopień trudności), a odpowiedź - ocenę. Aktualna ocena testu jest znormalizowaną średnią ważoną ocen udzielonych odpowiedzi. Niezależnie ocenianie może być prowadzone w co najwyżej 9 aspektach, a ocena ogólna jest sumą ocen w tych aspektach.

\* Każdy problem-pytanie ma indywidualny (paragrafowy) numer. Wybranie następnego - kolejnego - pytania może być zrealizowane poprzez:

- podanie jego numeru,
- podanie paragrafu (wybór losowy ze zbioru pytań),
- podanie wagi pytania,
- podanie paragrafu i wagi pytania.

Wskazanie następnego pytania wg powyższych reguł może być warunkowe. W warunkach logicznych mogą występować aktualnie: ocena ogólna, oceny w aspektach, ilość zadanych pytań, sumy wag zadanych pytań. Brak określenia następnego pytania oznacza koniec sesji.

\* Po zakończeniu sesji podawana jest ocena ogólna, oceny w poszczególnych aspektach oraz najlepsze osiągnięte dotychczas przez egzaminowanego wyniki i najlepsze wyniki osiągnięte przez wszystkich egzaminowanych tym testem. Prowadzony jest zapis wyników wszystkich egzaminów.

\* Każdy test zapisany jest w postaci zbioru ASCII o prostej składni.

Własności systemu umożliwiają komputerową realizację różnorodnych sposobów i form egzaminowania i nauczania.

*Dokumentacja systemu: "Opis systemu JKSOPW" oraz "Tutorial".*

*Zapewniamy bezpłatne konsultacje w 12-miesięcznym okresie gwarancyjnym.*

*Pomagamy - na specjalne życzenie - w opracowywaniu testów i przeprowadzeniu egzaminu w laboratoriach Instytutu Maszyn Matematycznych.*



# INSTYTUT MASZYN MATEMATYCZNYCH



OFERUJE

## KURSY KOMPUTEROWE

poranne, popołudniowe, sobotnie,  
dla początkujących PODSTAWOWE i UKIERUNKOWANE,  
dla zaawansowanych SPECJALISTYCZNE,  
*prowadzone przez doświadczonych specjalistów  
w nowoczesnych laboratoriach IMM,*  
gdzie każdy słuchacz pracuje na osobistym komputerze  
dostępnym do ćwiczeń także poza godzinami zajęć  
i otrzymuje nieodpłatnie materiały szkoleniowe.

### SZKOLIMY W NASTĘPUJĄCYM ZAKRESIE:

System operacyjny DOS, WORD, WORDPERFECT, DRAWPERFECT, QR-TEKST,  
LOTUS, QUATTRO-PRO, BTRIEVE, DBASE, CLIPPER, AUTOCAD;  
środowisko WINDOWS, WORD FOR WINDOWS, QR-TEKST FOR WINDOWS,  
AMI-PRO, WORKS FOR WINDOWS, QUATRO-PRO FOR WINDOWS,  
EXCEL, ACCESS, CORELDRAW, PAGEMAKER;  
System operacyjny UNIX - podstawy, użytkowanie, zarządzanie;  
Sieci lokalne - NOVELL, WINDOWS FOR WORKGROUPS, poczta elektroniczna;  
KOMPUTER DLA KSIĘGOWYCH, PC SERVIS, JĘZYK C++;  
KOMPUTER W BIURZE POSELSKIM.

Również inne kursy na zamówienie. Istnieje możliwość uzgodnienia zakresu i terminu,  
a także przeprowadzenia szkolenia u klienta.

Polecamy przeprowadzone już w licznych przypadkach  
**KURSY DLA KIEROWNICTWA FIRMY.**

Oplata za 45-godzinny, 2-tygodniowy kurs podstawowy - 2 600 000 zł.

Instytut prowadzi DORADZTWO KOMPUTEROWE, wykonuje EKSPERTYZY  
i inne OPRACOWANIA DOTYCZĄCE KOMPUTERYZACJI  
I ZASTOSOWAŃ INFORMATYKI W PRZEDSIĘBIORSTWACH,  
przyjmuje zlecenia na OPRACOWANIE OPROGRAMOWANIA.

Informacja:

ul. Krzywickiego 34, 02-078 Warszawa,  
fax 299270, tlx 817880, tel 299164 lub 6218441 w. 500  
e-mail [imasmat@frodo.nask.org.pl](mailto:imasmat@frodo.nask.org.pl)





## INSTYTUT MASZYN MATEMATYCZNYCH



*Oferujemy*

### KOMPUTEROWY SYSTEM KONTROLI DOSTĘPU I REJESTRACJI CZASU PRACY SKR

SKR identyfikuje karty magnetyczne, prowadzi ewidencję czasu pracy pracowników, a także może generować informacje dla komputerowych systemów obliczania plac. System składa się z minikomputera IBM PC oraz jednego lub wielu czytników kart magnetycznych (do 32) i oprogramowania CHRONOS działającego pod systemem DOS w wersji 3.10 lub nowszej.

Czytnik kart magnetycznych AS-1400:

- zasilanie +5V lub +12V (zasilacz wchodzi w skład wyposażenia),
- karty magnetyczne - wg ISO 2894, ścieżka 2,
- klawiatura dwunastoznakowa,
- przesuw ręczny,
- długość kodu - do 40 znaków,
- pamięć do 5000 rejestracji we/wy,
- temperatura pracy: +5 do +45 °C,
- żywotność głowicy do 1 mln odczytów.

CHRONOS w wersji standardowej pozwala:

- redagować dokumenty opisujące pracowników (do 2000), usuwać je, modyfikować i drukować,
- definiować parametry dotyczące pracowników, w tym czas pracy, różne statusy i in.,
- ustawiać czas i datę w komputerze i czytnikach,
- programować czytnik, np. wprowadzać i usuwać numery kart akceptowanych,
- rejestrować wejścia i wyjścia pracowników i osób obcych (zaproszonych),
- przeszukiwać informacje o poruszaniu się pracowników, np. uzyskać informacje o pracownikach, którzy w określonym czasie wychodzili z zakładu,
- generować informacje dla istniejących w przedsiębiorstwie systemów kadrowo-placowych,
- drukować na drukarce zestawienia o wejściach i wyjściach pracowników, raporty miesięczne, dzienne i inne.

*Ten najtańszy system rejestracji i kontroli czasu pracy  
ułatwi pracę Twoim pracownikom .*



## INSTYTUT MASZYN MATEMATYCZNYCH



*Jeśli:*

- *projektujecie lub uruchamiacie systemy mikroprocesorowe,*
  - *musicie zaprogramować układ scalony,*
  - *przewadzicie pomiary wielkości elektrycznych*
- wykorzystajcie oferowane przez Instytut profesjonalne narzędzia*

### MIKROPROCESOROWY SYSTEM WSPOMAGANIA PROJEKTOWANIA MSWP-92

(współpracujący z minikomputerami IBM PC)

- \* emulatory układów procesorów 8080, Z80, 8085, 8086/88, 80286, 8035/39/40/48/49/50, 8031/51/52/44/154/652/851, 80515/535, 80C552/562
- \* programator EP-11 dla pamięci EPROM i struktur logicznych PAL
- \* uniwersalny programator/tester UP-1
- \* analizator stanów logicznych ASL-24
- \* symulator pamięci stałych SYM-11
- \* lampa kasująca EE-12

### SYSTEM POMIAROWY VIRT II

(karty do minikomputera IBM PC)

- \* RC681 - mostek RLC
- \* AD611B - przetwornik 12-bitowy A/C, C/A
- \* IO601A - cyfrowe we/wy
- \* WD671A - rejestrator przebiegów
- \* PG621A - generator przebiegów
- \* ROM-DISK - zastępuje mechanizm dyskowy "A"

*Serwis oraz bezpłatne konsultacje w 12-miesięcznym okresie gwarancyjnym.*

*Pełna dokumentacja eksploatacyjna. Przystępne ceny.*

*Możliwość odpłatnego wypożyczenia.*

**ZAPRASZAMY**



BIBLIOTEKA GŁÓWNA  
Politechniki Śląskiej

P. 3057/94