

Adam SCHMIDT

WYBRANE METODY OPISU JEDNOMAGISTRALOWEGO SYSTEMU WIELOPROCESOROWEGO I ICH WERYFIKACJA PRAKTYCZNA

Streszczenie. W pracy przedstawiono kilka teoretycznych metod opisu jednomagistralowego systemu wieloprocesorowego. Zaproponowano udoskonalony opis, który uwzględni rzeczywiste parametry, obliczone na bazie istniejącego komputera wielomikroprocesorowego. Porównano wprowadzony model z teoretycznymi, omówiono różnice. Zamieszczono wyniki eksperymentalnego badania systemu i stwierdzono dobrą zgodność zaproponowanego modelu z doświadczeniem.

1. WSTĘP

Bardzo szybki postęp w dziedzinie VLSI spowodował wzrost zainteresowania mikroprocesorami jako podstawą systemu wieloprocesorowego. Dążenie do zmniejszenia kosztów skierowało z kolei uwagę konstruktorów w kierunku systemów magistralowych. Istnieją w tej dziedzinie sprawdzone rozwiązania, oparte w dużej mierze na standardowych magistralach, np. MULTIBUS [22] i VMEbus [7]. Dla wielu komputerów opracowano też oryginalną magistralę: [4], [23], [27], [21], [13], [8].

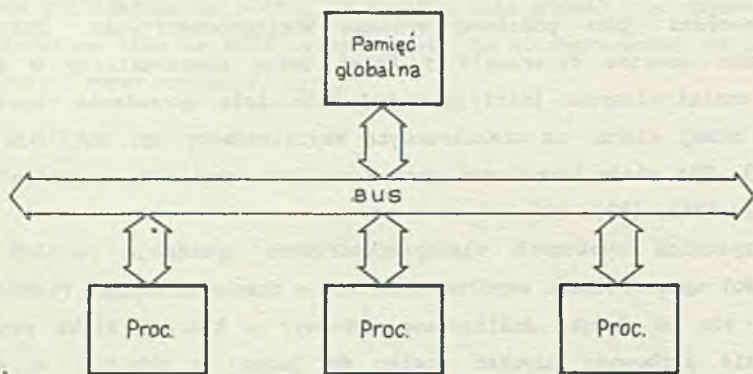
We wszystkich systemach wieloprocesorowych występuje problem spadku efektywności spowodowanego współzawodnictwem o zasoby globalne. Problemem tym zajmowano się od dawna. Analizowano systemy, w których kilka procesorów jednocześnie próbowało uzyskać dostęp do jednej z pamięci, w systemie złożonym z N procesorów i P pamięci, połączonych łącznicą krzyżową (crossbar switch) [5] lub magistralą [20]. Rozpatrywano też strukturę komunikacji między procesorami [11] i procesami [19]. Całościowe ujęcie problemu zawiera na przykład książka [3].

Podawano także wyniki eksperymentalnego badania efektywności w istniejących systemach. Odnośnie architektury magistralowej można tu wymienić [4], [23], [3], [13].

Niniejsza praca ma na celu opis teoretyczny efektywności istniejącego systemu wieloprocessorowego na przykładzie komputera [26]. W tym celu przedstawione zostaną modele matematyczne dla różnych uproszczeń, które następnie porównane zostaną z modelem bardziej szczegółowym, uwzględniającym rzeczywiste parametry. Na końcu zamieszczone zostaną wyniki badania eksperymentalnego.

2. PARAMETRY SYSTEMU WIELOPROCESOROWEGO

Jednomagistralowy system wieloprocessorowy (rys. 1) składa się w ogólnym przypadku z N procesorów i pamięci globalnej, połączonych jedną magistralą. W opisywanym komputerze każdy procesor posiada ponadto pamięć lokalną, w której przechowywany jest program i częściej używane (lokalne) dane. Procesor odwołuje się do pamięci globalnej tylko w celu pobrania kolejnej porcji danych do przetwarzania. Pomiedzy dwoma kolejnymi odwołaniami do pamięci upływa czas, którego długość jest zmienną losową o skomplikowanym na ogół rozkładzie, i o wartości średniej t_z . Czas zajęcia magistrali przez jeden procesor jest natomiast stały i wynosi t_m .



Rys. 1. Jednomagistralowy system wieloprocessorowy

Fig. 1. Single bus multiprocessor system

Każdy procesor może znajdować się w trzech stanach: pracy lokalnej (komunikuje się tylko z pamięcią lokalną), oczekiwania (gdy odwołuje się do pamięci globalnej, która jest przyznana innemu procesorowi) i transmisji (wykonuje cykl zapisu do lub odczytu z pamięci globalnej).

W celu opisanía własności systemu wieloprocesorowego zaproponować można kilka parametrów. Najszerzej używanym jest współczynnik przyspieszenia S_p (speedup factor, effective CPU, processing power, system throughput); wskazujący, o ile szybciej przebiegają obliczenia w systemie wieloprocesorowym w porównaniu z jednoprocessorowym. Parametrem pochodnym jest efektywność (efficiency) U , wyrażająca spadek prędkości jednego procesora na skutek konfliktów. Jest oczywiście $S_p = N \cdot U$. Efektywność można też wyrazić inaczej jako stosunek liczby cykli aktywnych procesora do wszystkich cykli. Należy jednak określić, które z cykli procesora są aktywne. Na ogół przyjmuje się, że aktywne są tylko cykle odpowiadające pracy lokalnej. Trafniejsze wydaje się jednak (zostanie to uwidocznione przez analizę rzeczywistego systemu) przyjęcie także cykli transmisji jako aktywnych. Taka klasyfikacja będzie stosowana w niniejszej pracy. Można zatem podać:

$$S_p = \frac{(\text{cykle pracy lokalnej} + \text{cykle transmisji}) \cdot N}{\text{cykle pracy lokalnej} + \text{cykle transmisji} + \text{cykle oczekiwania}} \quad (2.1)$$

Zastosowany model systemu wieloprocesorowego zależy jest od założonego rozkładu czasu aktywnego (między odwołaniami) i czasu transmisji, a także od sposobu wyboru przez układ arbitrażu następnego procesora do komunikacji z magistralą. Ponieważ rozkład czasu między odwołaniami zależy jest od wykonywanego programu, przyjmuje się na ogół rozkłady możliwie proste do analizy. W celu dalszego jej uproszczenia przyjmuje się też inne niż deterministyczny (stały) rozkłady czasu transmisji. Parametry te można zapisać w uproszczonej notacji Kendalla [3]: $Z/T/KOL$, gdzie Z jest rozkładem czasu między zgłoszeniami, T - czasu transmisji, a KOL - charakterem arbitrażu (w teorii masowej obsługi zwanym regulaminem kolejki). Jako krytyczne przyjęto rozkłady obsługi: wykładowy M (najmniej efektywny) i stały D (najbardziej). Dlatego modele matematyczne uwzględniające te rozkłady zostaną opisane kolejno w rozdz. 3.2 do 3.4 (3.2 - wykładowy, 3.3 i 3.4 - stały). Pomimo, że model ze stałym czasem obsługi wydaje się być najbardziej adekwatny, to jednak model z wykładowym jest prostszy i pozwala uzyskać rozkład prawdopodobieństwa liczby procesorów oczekujących, co okaże się użyteczne przy analizie rzeczywistego systemu. Jeżeli chodzi o regulamin kolejki to uwzględnione zostaną: naturalny (First Come First Served FCFS) i priorytetowy (Fixed Priorities FP). Szeroki wybór modeli znaleźć można w bogatej literaturze, np.: [15], [3], [29], [17]. W rozdz. 3.1 podane zostaną proste wzory na maksymalną i minimalną efektywność, niezależne od parametrów.

3. MODELE MATEMATYCZNE SYSTEMU

3.1. Graniczne wartości współczynnika przyspieszenia

W pracy [25] można znaleźć funkcje wiążące współczynnik przyspieszenia z liczbą procesorów dla najgorszego i najlepszego przypadku. Wyznaczają one obszar, w którym musi się znaleźć wspomniana zależność dla konkretnej metody opisu. Dla uproszczenia założono pełną synchroniczność systemu, tj. że zgłoszenia mogą się pojawiać tylko w określonych chwilach czasu.

W celu obliczenia najgorszego przypadku przyjęto, że zgłaszający się procesor wpada w konflikt ze wszystkimi pozostałymi procesorami. Jest wtedy:

$$Sp_{\min} = \frac{tz + tm}{tz + tm + (N-1) \cdot tm} \cdot N \quad (3.1)$$

Jeżeli chodzi o najlepszy przypadek, to zachodzi on dla procesorów w pełni zsynchronizowanych, tj. takich, dla których czas pracy lokalnej między dwoma zgłoszeniami jest stały. Wtedy $Sp=N$ tak długo, jak długo magistrala nie jest w pełni wykorzystana. Podany w [25] warunek całkowitego wykorzystania magistrali można rozszerzyć na systemy o dowolnym rozkładzie czasów pracy aktywnej i transmisji.

Przy maksymalnym obciążeniu na jeden cykl transmisji o średniej długości tm przypada czas aktywny równy średnio $tm+tz$; jest więc:

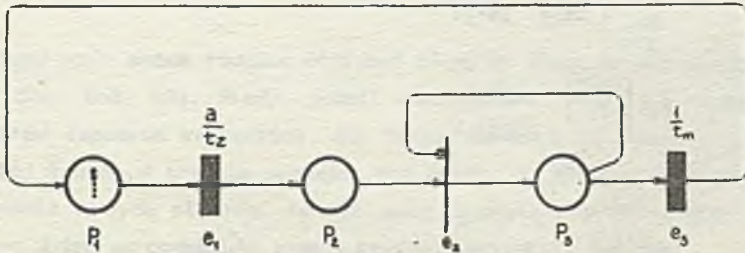
$$\lim_{N \rightarrow \infty} Sp = \frac{tz}{tm} + 1 \quad (3.2)$$

3.2. Obliczenie współczynnika przyspieszenia dla wykładniczego czasu obsługi (systemy M/M/- i G/M/FCFS) na podstawie teorii masowej obsługi

Jako punkt wyjścia do analizy można rozpatrzyć przypadek, gdy system składa się tylko z 2 procesorów (dla jednego procesora jest oczywiście $Sp=1$). Obserwując teraz jeden z nich można stwierdzić, że spadek efektywności wystąpi w przypadku wygenerowania przezeń zgłoszenia wtedy, gdy magistralę zajmuje drugi procesor. Prawdopodobieństwo takiego zdarzenia (konfliktu) P_{konf} jest równe:

$$P_{\text{konf}} = \frac{tm}{tm + tz} \quad (3.3)$$

Rozszerzając to rozumowanie można podać [9], że liczba procesorów, które napotyka procesor obserwowany zgłaszając się do magistrali jest równa liczbie tychże, dokonujących transmisji bądź czekających, w systemie złożonym z $N-1$ procesorów. Aby obliczyć tę liczbę można sięgnąć do teorii masowej obsługi, dokonując od razu założeń, co do rozkładów czasów i charakteru arbitrażu. Najprostszy do opisu jest układ o wykładniczym czasie obsługi i między zgłoszeniami, przy dowolnym regulaminie kolejki. Jego działanie można przedstawić prosto przy użyciu uogólnionych stochastycznych sieci Petri (GSPN) [3], jak to zrobiono na rys. 2. W sieci tej każdy znacznik reprezentuje jeden procesor spośród wszystkich M , tranzycja e_2 jest natychmiastowa, zaś tranzycje e_1 i e_3 są czasowe (timed transitions), o współczynnikach odpalania odpowiednio a/tz i $1/t_m$, gdzie a jest liczbą znaczników w miejscu P_1 . Stałe $1/tz$ i $1/t_m$ są jednocześnie parametrami rozkładów wykładniczych odpowiednio czasów między zgłoszeniami i obsługi. Poszczególne miejsca odpowiadają stanom, w których może znajdować się każdy procesor.



Rys. 2. Działanie systemu wieloprocessorowego - schemat GSPN:

P_1 - procesory w trakcie pracy lokalnej, P_2 - procesory czekające,

P_3 - procesory w trakcie dostępu do pamięci globalnej

Fig. 2. Generalized stochastic Petri net diagram for a single bus multiprocessor system:

P_1 - local active, P_2 - waiting, P_3 - accessing the bus

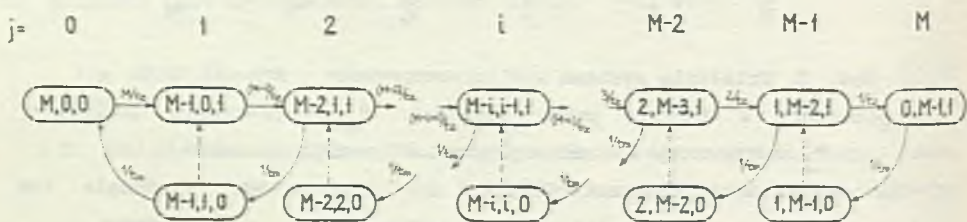
Na podstawie podanej sieci można napisać graf przejść między stanami układu (rys. 3). Każdy z nich reprezentowany jest przez trójkę $\langle a, w, m \rangle$, gdzie a jest liczbą procesorów w trakcie pracy lokalnej, w - oczekujących na przydzielenie magistrali, zaś m - będących w trakcie transmisji. Ponieważ w grafie interesujące są tylko przejścia czasowe (oznaczone ciągłą strzałką), można go uprościć eliminując m . Ponadto $a+w+m=M$, czyli stan systemu da się opisać tylko jedną zmienną: $j=w+m$, oznaczającą liczbę procesorów, które

zgłosiły się do magistrali globalnej. Aby obliczyć teraz prawdopodobieństwa stanów P_j należy zbilansować wejścia i wyjścia dla każdego stanu, otrzymując znane równania życia i śmierci tzw. modelu konserwatora [17], [18], [10]:

$$\left\{ \begin{array}{l} p_1/t_m = p_0 \cdot M/t_z \\ p_j \cdot (1/t_m + (M-j)/t_z) = p_{j-1} \cdot (M-j+1)/t_z + p_{j+1}/t_m \\ \sum_{j=0}^M p_j = 1 \end{array} \right. \quad (3.4)$$

Iteracyjne rozwiązywanie takiego układu równań prowadzi do wyniku:

$$\left\{ \begin{array}{l} p_0 = \frac{1}{\sum_{j=0}^M \left(\frac{t_m}{t_z} \right)^j \frac{M!}{(M-j)!}} \\ p_j = p_0 \left(\frac{t_m}{t_z} \right)^j \frac{M!}{(M-j)!} \end{array} \right. \quad (3.5)$$



Rys. 3. Graf przejść między stanami systemu wieloprocessorowego $\langle a, w, m \rangle$:

a - liczba procesorów w trakcie pracy lokalnej, w - liczba procesorów oczekujących, m - liczba procesorów w trakcie dostępu do pamięci globalnej

Fig. 3. State transitions rate for the system; states are described as

$\langle a, w, m \rangle$, where:

a - local active processors, w - waiting (queued) processors, m - processors accessing the bus

Można już zatem obliczyć średnią liczbę procesorów L zastanych przez zgłaszający się:

$$L = \sum_{j=0}^{N-1} j \cdot p_j^{(N-1)} \quad (3.6)$$

gdzie $p_j^{(N-1)}$ oznacza prawdopodobieństwo obliczone ze wzorów (3.5) dla $M=N-1$.

Średni czas oczekiwania W na zakończenie transmisji przez wszystkie procesory wynosi:

$$W = L \cdot t_m = \sum_{j=0}^{N-1} j \cdot p_j^{(N-1)} \cdot t_m \quad (3.7)$$

W związku z tym szukany współczynnik przyspieszenia będzie równy:

$$S_p = \frac{t_z + t_m}{t_z + t_m + L \cdot t_m} \cdot N \quad (3.8)$$

Tożsamy wzór można również otrzymać stosując model konserwatora (równania (3.5)) dla $M=N$, [3]. Wtedy, jeżeli oznaczy się przez L' średnią liczbę procesorów (spośród wszystkich N), które oczekują na dostęp do magistrali globalnej lub są w trakcie dostępu, zaś przez D średnią sumaryczną długość przebywania w tych stanach, to dla schematycznie przedstawionego na rys. 4 diagramu można na podstawie zasady równości strumieni napisać:

$$\frac{N-L'}{t_z} = \frac{1-p_0^{(N)}}{t_m} \quad (3.9)$$

gdzie:

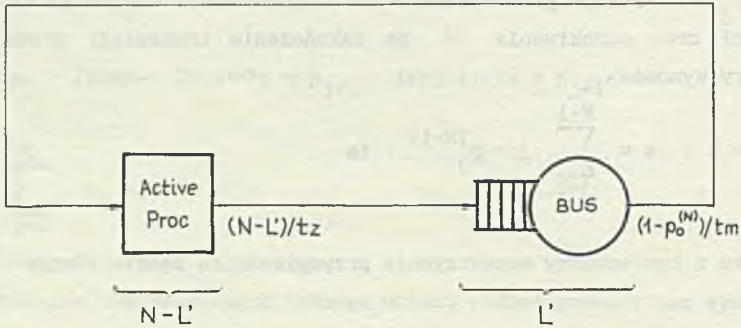
$$L' = \sum_{j=0}^N j \cdot p_j^{(N)} \quad (3.10)$$

zaś $(1-p_0^{(N)})$ wyraża prawdopodobieństwo, że magistrala jest w ogóle zajęta. Z twierdzenia Little'a wynika:

$$D = \frac{L' \cdot t_z}{(N-L')} \quad (3.11)$$

zatem:

$$D = \frac{L' \cdot t_m}{(1-p_0^{(N)})} \quad (3.12)$$



Rys. 4. Schemat kolejkowy systemu wieloprocessorowego

Fig. 4. A queuing model for the system

Czas cyklu t_c (od wygenerowania zgłoszenia do wygenerowania następnego) wynosi:

$$t_c = t_z + D \quad (3.13)$$

z którego na czas aktywny przypada $t_z + t_m$.

Współczynnik przyspieszenia jest więc równy:

$$S_p = \frac{t_z + t_m}{t_z + D} \cdot N \quad (3.14)$$

czyli:

$$S_p = \frac{t_z + t_m}{t_z + \frac{L'}{(1-p_0)} \cdot t_m} \cdot N \quad (3.15)$$

Jeżeli ograniczyć rozważania do systemów o naturalnym regulaminie kolejki (FCFS), to otrzymane wyniki będą ważne także w przypadku dowolnego czasu między zgłoszeniami, czyli do G/M/FCFS. Otrzymane bowiem na podstawie analizy wartości średniej równania [3] na p_j dla M procesorów mają postać:

$$p_j = \frac{1}{G} \cdot j! \cdot (tm)^j \cdot \sum_{m \in M_{we}} \prod_{k=1}^M (tz)^{m_k} \quad (3.16)$$

gdzie G jest współczynnikiem normalizacji wyrażającym $\sum_{j=0}^M p_j = 1$ i wynoszącym:

$$G = \sum_{j=0}^M j! \cdot (tm)^j \cdot \sum_{m \in M_{we}} \prod_{k=1}^M (tz)^{m_k} \quad (3.17)$$

M_{we} zaś jest zbiorem wektorów $m = (m_1, m_2, \dots, m_N)$ takich, że $m_j = 0$ lub $m_j = 1$

i $\sum_{i=1}^M m_i = M-j$; sumowanie po $m \in M_{we}$ wyraża przy danym j obliczenie

sumy dla wszystkich możliwych kombinacji procesorów.

Biorąc pod uwagę, że $\sum_{i=1}^M (1-m_i) = j$, oraz że

$$\sum_{m \in M_{we}} 1 = \frac{M!}{(M-j)! \cdot j!} \quad (3.18)$$

otrzymujemy:

$$p_j = \frac{1}{G} \cdot \frac{M!}{(M-j)!} \cdot \left(\frac{tm}{tz} \right)^j \cdot (tz)^M \quad (3.19)$$

gdzie

$$G = (tz)^M \cdot \sum_{j=0}^M \frac{M!}{(M-j)!} \cdot \left(\frac{tm}{tz} \right)^j \quad (3.20)$$

czyli znane wzory (3.4), w których $p = G/tz_0^M$.

3.3. Obliczenie współczynnika przyspieszenia dla układu ze stałym czasem obsługi i o sztywnych priorytetach (G/D/FP)

W artykule [12] zakładając synchroniczny charakter magistrali i priorytetowy regulamin kolejki obliczono prawdopodobieństwo, że w danym cyklu procesora wystąpi konflikt z procesorem o wyższym priorytecie. Taki konflikt oznacza oczywiście oczekiwanie i związany z nim spadek efektywności.

U podstaw tej analizy leży spostrzeżenie, że konflikt może być skutkiem albo odwołania się w danym cyklu do zajętej pamięci, albo trwającego od k cykli oczekiwania. Dlatego całkowite jego prawdopodobieństwo jest sumą niezależnych prawdopodobieństw warunkowych konfliktów trwających od $1, 2, \dots, k, \dots, \infty$ cykli. Znając prawdopodobieństwo $b = tm/(tz+tm)$, że pojedynczy procesor odwoła się do magistrali można dla j -tego procesora iteracyjnie obliczyć prawdopodobieństwo konfliktu, który trwa dokładnie od k cykli (t.j. pod warunkiem, że nie było konfliktu w $t-k-1$ cyklu): $p_c^{(j)}(k)$; oraz prawdopodobieństwo, że co najmniej 1 spośród $j-1$ procesorów o wyższym priorytecie zajmuje magistralę: $p_s^{(j-1)}$. Można napisać, że:

$$p_c^{(j)}(k) = (1-p_c^{(j)}) \cdot p_s^{(j-1)} \cdot b \cdot (p_s^{j-1})^k \quad (3.21)$$

gdzie pierwszy czynnik jest prawdopodobieństwem tego, że nie było konfliktu przed $k+1$ cyklami, drugi - że przy danym zgłoszeniu magistrala była zajęta, trzeci - że dany procesor się zgłosił, czwarty - że magistrala jest zajęta od k cykli. Suma ciągu tych prawdopodobieństw w granicy dla k dążącego do ∞ jest szukanym $p_c^{(j)}$:

$$p_c^{(j)} = \frac{1}{1 + \frac{1-p_s^{(j-1)}}{p_s^{(j-1)} \cdot b}} \quad (3.22)$$

Z kolei prawdopodobieństwo zajęcia magistrali przez co najmniej 1 procesor spośród $j-1$ wyraża wzór:

$$p_s^{(1)} = b \quad (3.23)$$

$$p_s^{(j-1)} = (1-p_c^{(j-1)}) \cdot (1-p_s^{(j-2)}) \cdot (1-b)$$

Wartości p_s i p_c ze wzorów (3.21) i (3.22) oblicza się na przemian dla kolejnych j , zakładając oczywiście $p_c^{(1)} = 0$. Współczynnik przyspieszenia można wtedy policzyć następująco:

$$Sp = \sum_{j=1}^N (1-p_c^{(j)}) \quad (3.24)$$

3.4. Opis systemu oparty na analizie stopnia wykorzystania pamięci globalnej (dla systemów G/D/-)

W artykułach [24], [6] poświęconych porównaniu efektywności kilku architektur komputerów wykorzystujących pamięć notatnikową (Cache) zaproponowano interesujący model opisujący synchroniczne systemy wielomikroprocesorowe z dzielonymi zasobami globalnymi, np. pamięciami czy magistralami. Model ten opiera się na spostrzeżeniu (na podstawie symulacji), że efektywność systemu zależy w przeważającej mierze od intensywności odwołań do pamięci (tj. iloczynu prawdopodobieństwa odwołania i długości transmisji) a nie od charakteru tych odwołań (tj. rozkładu każdej z tych wielkości z osobna). Prawdziwość tego spostrzeżenia potwierdza przeprowadzona w rozdz. 3.2 analiza (intensywność odwołań była tam zależna od ilorazu t_m/t_z). Opierając się na nim można przyjąć uproszczenie, że cykle związane z dostępem do pamięci (o długości $t_m + L \cdot t_m$) nie są zgrupowane razem, lecz rozproszone równomiernie w czasie. Następnie porównuje się obciążenie m wnoszone przez jeden procesor (inaczej prawdopodobieństwo zajęcia magistrali) z obciążeniem, jakie może przyjąć magistrala. Obciążenie m można wyliczyć jako:

$$m = \frac{t_m + L \cdot t_m}{t_z + t_m + L \cdot t_m} \quad (3.25)$$

gdzie t_m jest stałym czasem cyklu wymiany, a t_z średnim odstępem między odwołaniami. Obciążenie pamięci B równe jest prawdopodobieństwu, że co najmniej jeden procesor wygeneruje zgłoszenie:

$$B = (1 - (1-m)^N) \quad (3.26)$$

Z drugiej strony obciążenie to jest równe średniej liczbie cyklów transmisji wykonywanych przez wszystkie procesory:

$$B = N \cdot \frac{t_m}{t_z + t_m + L \cdot t_m} \quad (3.27)$$

Porównując wzory (3.25) i (3.26) otrzymać można równanie z jedną niewiadomą L :

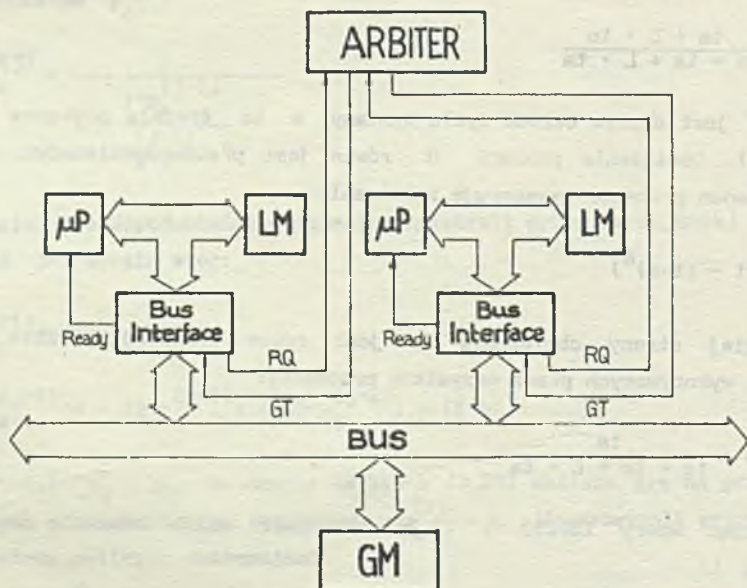
$$\left(1 - \left(1 - \frac{t_m + L \cdot t_m}{t_z + t_m + L \cdot t_m}\right)^N\right) - N \cdot \frac{t_m}{t_z + t_m + L \cdot t_m} = 0 \quad (3.28)$$

które można rozwiązać numerycznie; jako wartość początkową przyjmując $L=0$. Obliczone L można następnie podstawić do równania (3.8), obliczając współczynnik przyspieszenia.

4. ANALIZA EFEKTYWNOŚCI UWZGLĘDNIAJĄCA ZALEŻNOŚCI CZASOWE W RZECZYWISTYM SYSTEMIE

W opartym na mikroprocesorach Intel 8085 komputerze, który posłużył do badań nad efektywnością, zastosowano buforowany układ dostępu do magistrali [26]. Dzięki niemu czas dostępu do magistrali nie jest zależny od wymagań mikroprocesora, a jedynie od szybkości tej pamięci. Wspomniany układ umożliwia ponadto zmienianie czasu transmisji do celów eksperymentalnych. Zasada działania polega na odcinaniu procesora po ustalonym czasie $t_{ARB} + t_{EXCH}$ od magistrali przy zapisie, i zatraskiwaniu pobranej z pamięci danej przy odczycie, tak że dostępna jest ona dla procesora dowolnie długo; w tym czasie magistrala może być wykorzystywana przez inne procesory.

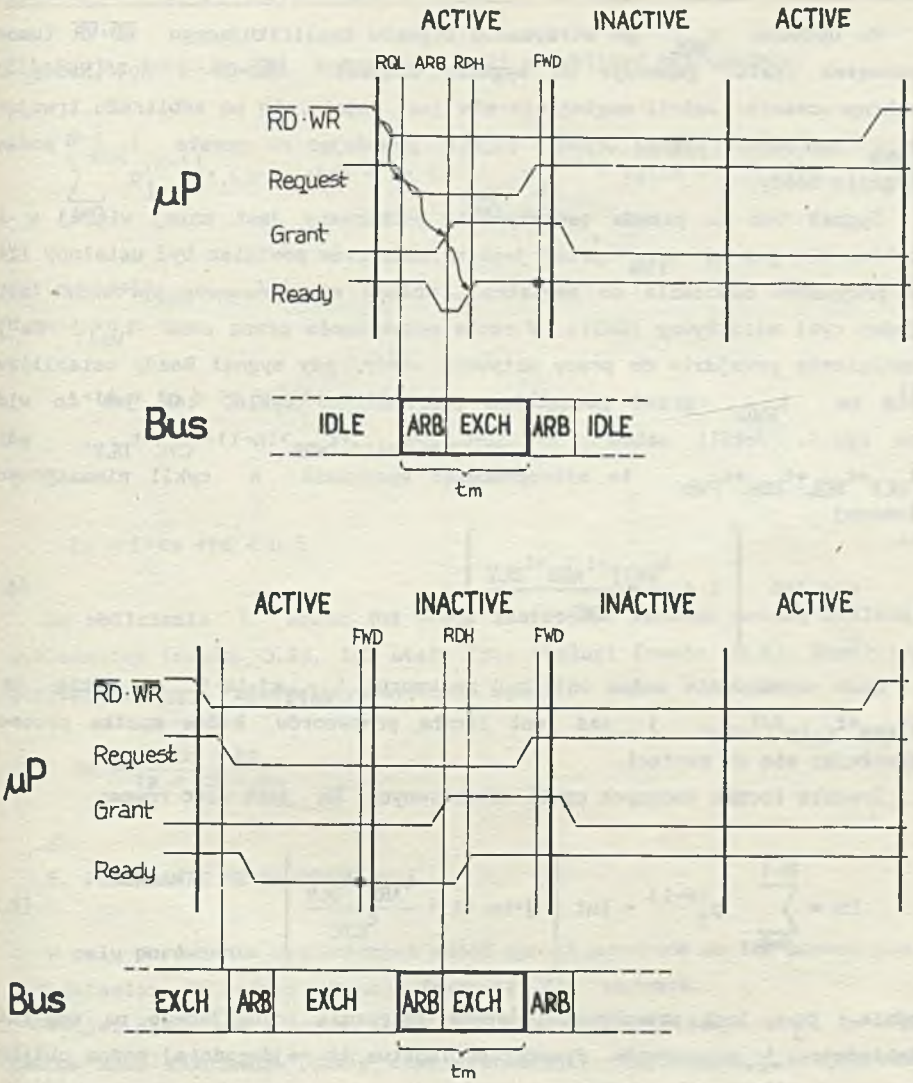
Uproszczone przebiegi czasowe układu współpracy z magistralą (którego schemat pokazany jest na rys. 5) przedstawia rys. 6.



Rys. 5. Komputer wielomikroprocesorowy z układem dostępu do magistrali:
LM - pamięć lokalna (local memory), GM - pamięć globalna (global memory)

Fig. 5. Existing multimicroprocessor system with the bus access unit:

LM - local memory, GM - global memory



Rys. 6. Przebiegi czasowe przy odwołaniu do wolnej (a) i zajętej (b) magistrali

Parametry: EXCH=150ns, ARB=70ns, RQL=40ns, FWD=20ns, RDH=15ns

Fig. 6. Timings for the bus access, when the bus is idle (a) and when the bus is occupied (b)

Parameters: EXCH=150ns, ARB=70ns, RQL=40ns, FWD=20ns, RDH=15ns

Po upływie t_{RQL} po otrzymaniu sygnału kwalifikującego $\overline{RD} \cdot \overline{WR}$ (umowny początek cyklu) generuje on sygnały Request ($RQ=0$) i Not Ready dla mikroprocesora. Jeżeli magistrala nie jest zajęta, to po arbitrażu trwającym t_{ARB} nadchodzi sygnał Grant ($GT=1$), powodując po czasie t_{RDH} podanie sygnału Ready.

Sygnał ten co prawda teoretycznie próbkowany jest mniej więcej w 1/3 cyklu, ale już od t_{FWD} przed jego rozpoczęciem powinien być ustalony [14]. W przypadku odwołania do magistrali wolnej mikroprocesor wprowadzi zatem jeden cykl nieaktywny (Wait). W razie oczekiwania przez czas t_{WAIT} na jej zwolnienie przejdzie do pracy aktywnej wtedy, gdy sygnał Ready ustabilizuje się na t_{FWD} przed początkiem poprzedniego cyklu, tak jak to widać na rys.6. Jeżeli zatem $n \cdot t_{CYC} - t_{DLY} > t_{WAIT} + t_{ARB} > (n-1) \cdot t_{CYC} - t_{DLY}$, gdzie $t_{DLY} = t_{RQL} + t_{RDH} + t_{FWD}$, to mikroprocesor wprowadzi n cykli nieaktywnych. Inaczej

$$n = \text{Int} \left(1 + \frac{t_{WAIT} + t_{ARB} + t_{DLY}}{t_{CYC}} \right) \quad (4.1)$$

Czas oczekiwania można obliczyć ze wzoru $t_{WAIT} = j \cdot t_m \cdot t_{CYC}$, gdzie $t_m = (t_{ARB} + t_{EXCH}) / t_{CYC}$, j zaś jest liczbą procesorów, które spotka procesor odwołując się do pamięci.

Średnia liczba dodanych cykli nieaktywnych E_n jest więc równa:

$$E_n = \sum_{j=0}^{N-1} p_j^{(N-1)} \cdot \text{Int} \left(j \cdot t_m + 1 + \frac{t_{ARB} + t_{DLY}}{t_{CYC}} \right) \quad (4.2)$$

gdzie p_j jest prawdopodobieństwem zdarzenia polegającego na spotkaniu dokładnie j procesorów. Prawdopodobieństwa te najdogodniej można obliczyć ze wzorów (3.4) dla $M=N-1$ i przyjmując, że czasy między zgłoszeniami i wymiany mają rozkłady wykładnicze o parametrach odpowiednio $1/t_z$ i $1/t_m$.

Porównując wzór (4.2) ze wzorem (3.6) można zauważyć, że różnica między nimi polega na wprowadzeniu funkcji Int i składnika $1 + (t_{ARB} + t_{DLY}) / t_{CYC}$. Ich znaczenie można intuicyjnie wyjaśnić: Funkcja Int wyraża skokowy charakter dodawanych opóźnień (tylko pełne cykle). Wystąpienie składnika $1 + (t_{ARB} + t_{DLY}) / t_{CYC}$ powodowane jest przez fakt, że magistrala przekazywana jest następnemu procesorowi z opóźnieniem (a nie natychmiast, jak to zakłada model teoretyczny, por. np [1]) oraz, że reakcja procesora na przyznanie magistrali również następuje z opóźnieniem.

Eliminując funkcję Int można wzór (4.2) przybliżyć następująco:

$$\sum_{j=0}^{N-1} p_j^{(N-1)} \cdot (j \cdot t_m + t_d) < E_n < \sum_{j=0}^{N-1} p_j^{(N-1)} \cdot (j \cdot t_m + 1 + t_d) \quad (4.3)$$

gdzie $t_d = (t_{\text{ARB}} + t_{\text{DLY}}) / t_{\text{CYC}}$

W związku z tym:

$$L \cdot t_m + t_d < E_n < L \cdot t_m + t_d + 1 \quad (4.4)$$

oraz

$$E_n \approx L \cdot t_m + t_d + 0.5 \quad (4.5)$$

Do obliczenia L można już wtedy zastosować zarówno metodę zakładającą wykładniczy (rozd. 3.2), lub stały czas obsługi (rozd. 3.4). Współczynnik przyspieszenia w rzeczywistym systemie wynosi:

$$S_p = \frac{t_m + t_z}{t_m + t_z + E_n} \quad (4.6)$$

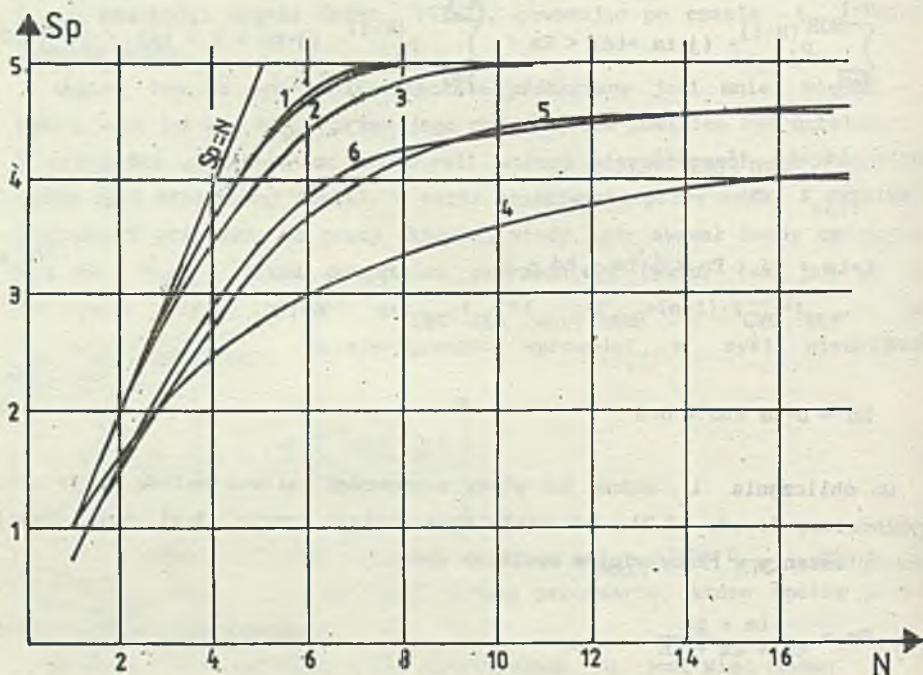
5. PORÓWNANIE METOD OPISU

W celu porównania wymienionych metod wyniki uzyskane za ich pomocą zostały przedstawione na jednym wykresie (rys. 7), dla $t_z/t_m = 4$.

Wyniki dla metod z rozdz. 3.3 i 3.4 są praktycznie takie same, ze względu na te same założenia (stały czas transmisji). Współczynnik przyspieszenia obliczony ze wzorów z rozdz. 3.2 (wykładniczy rozkład czasu obsługi) jest gorszy, ale różnice są niewielkie.

Podobnie dla krzywych opisujących współczynniki przyspieszenia dla rzeczywistego systemu - zmiana metody opisu nie zmienia w sposób istotny wyglądu funkcji.

Wyrażna jest natomiast różnica między przewidywaniami modelowymi a rzeczywistymi - sięga ona 20%.



Rys. 7. Przebiegi $Sp(N)$ dla różnych modeli:

1 - G/D/FP z rozdz.3.3, 2 - G/D/- z rozdz. 3.4, 3 - M/M/- z rozdz. 3.2,
 4 - najgorszy przypadek z rozdz. 3.1, 5 - system rzeczywisty, E_n ze wzoru
 (4.2), $p_j^{(N-1)}$ ze wzoru (3.5), 6 - system rzeczywisty, E_n ze wzoru (4.5),
 L ze wzoru (3.28)

Fig. 7. Speedup factor Sp versus number of processors N for different models:
 1 - G/D/FP from par. 3.3, 2 - G/D/- from par. 3.4, 3 M/M/- from par. 3.2,
 4 - worst case from par. 3.1, 5 - existing system, E_n from formula (4.2),
 $p_j^{(N-1)}$ from formula (3.5), 6 - existing system, E_n from formula (4.5),
 L from formula (3.28)

6. BADANIE EKSPERYMENTALNE SYSTEMU WIELOMIKROPROCESOROWEGO

W celu porównania przewidzianej teoretycznie efektywności systemu z faktyczną został napisany specjalny program eksperymentalny. Składa się on z pętli z jednym odwołaniem do pamięci globalnej, którego wykonanie uzależnione

jest od liczby bitów w bajcie pobranym z tablicy liczb pseudoprzyypadkowych. Gęstość odwołań w programie odpowiada mniej więcej programom transmisji blokowej, najbardziej obciążającym magistralę.

Odstępy między odwołaniami można wyliczyć analizując program lub zmierzyć za pomocą układu badania efektywności [SCH--], który zlicza cykle procesora CLK, odwołania do pamięci globalnej REQUEST i taktów nieaktywnych WAIT. Jest wtedy: $t_z = \text{CLK} / \text{REQUEST} - t_m$. Wyniki eksperymentów dla liczby procesorów $N=1$ i 2 porównano z wynikami teoretycznymi w tabeli 1.

Tabela 1

Porównanie teoretycznych i doświadczalnych wartości liczby taktów nieaktywnych E_n i współczynnika przyspieszenia S_p . W wartości E_n dla wyników teoretycznych D i E uwzględniono jeden takt nieaktywny dodawany przy każdym odwołaniu do pamięci globalnej

Typ		A	B	C	D	E	F
Określ. param.	eksper.	dokł. M/M/-	średni M/M/-	średni M/D/-	teoret. M/M/1	teoret. M/D/-	teoret. M/D/FP
Wykorz. wzory		4.2 1 3.4	4.5 1 3.5	4.5 1 3.27	3.6	3.27	3.23
N=2, E_n :	1.0120	1.0137	0.9626	0.9579	1.0094	1.0048	---
N=3, E_n :	1.0256	1.0275	0.9723	0.9628	1.0191	1.0097	---
N=2, S_p :	1.9603	1.9603	1.9623	1.9624	1.9604	1.9606	1.9998
N=3, S_p :	2.9398	2.9397	2.9429	2.9434	2.9401	2.9407	2.9994

Widoczna jest duża zgodność wyników eksperymentu, szczególnie z modelem A. Błąd E_n wynoszący 6.9% dla $N=3$ (i to po odliczeniu dodawanego zawsze taktu nieaktywnego) należy uznać za niewielki. Na dokładność E_n , inaczej niż na S_p , nie ma wpływu fakt małego obciążenia magistrali.

7. ZAKOŃCZENIE

Porównanie różnych metod opisu jednomagistralowych systemów wieloprocesorowych pozwala wysnuć wniosek, że modele te dają bardzo zbliżone rezultaty mimo różnych założeń i diametralnie różnych metod obliczeniowych. Włączenie do modelu niektórych parametrów rzeczywistych, jak całkowita liczba dodawanych cykli nieaktywnych, czas potrzebny na arbitraż i opóźniona reakcja mikroprocesora na przyznanie magistrali powoduje natomiast zasadnicze zmiany. Należy zatem wysnuć wniosek, że to parametry rzeczywiste systemu, na ogół zaniedbywane, mają decydujący wpływ na efektywność, komplikowanie zaś modelu matematycznego (w sensie zmian parametrów Z/T/KOL) nie daje istotnego zwiększenia dokładności opisu. Doświadczenie wniosek ten potwierdza. Wprowadzone parametry rzeczywiste mają tu zresztą sens ogólny i występują we wszystkich realizacjach opartych na mikroprocesorach. Przy okazji widać też wyraźnie, że nawet przy tak niekorzystnym obciążeniu magistrali jak występujące przy transmisji blokowej, ma ona jeszcze spore rezerwy. W przypadku architektury z pamięcią lokalną, takiej jak w [SCH--], nasycenie wynosi: $Sp_{\max} = 72.8$.

LITERATURA

- [1] M.Ajmone Marsan, M.Gerla: Markov Models for Multiple Bus Multiprocessor Systems, IEEE Trans. Comp., No.3, March 1982.
- [2] M.Ajmone Marsan, G.Balbo, G.Conte: Comparative Performance Analysis of Single Bus Multiprocessor Architectures, IEEE Trans. Comp., No.12, Dec. 1982.
- [3] M.Ajmone Marsan, G.Balbo, G.Conte: Performance Models of Multiprocessor Systems, The MIT Press, 1986.
- [4] J.W. Beyers, E.R.Zeller, S.D.Seccombe: VLSI Technology Packs 32-Bit Computer System into Small Package, HP Journal, August 1983.
- [5] D.P.Bhandarkar: Analysis of Memory Interference in Multiprocessors, IEEE Trans. Comp., No.9, Sept. 1975.
- [6] A. Bode, R.Q. Feitosa: Bewertung von Seicher - Konzepten fuer Multiprozessorsysteme, Informationstechnik it, 30(1988)2.
- [7] R.Brause: Prozessoren tauschen Nachrichten ueber Dual - Ported RAMs aus, VMEbus, April 1988.

- [8] F.Cesarini, G.Soda, G.Zappa: A Multiprocessor Architecture and its Application to Adaptive Control, Microprocessing and Microprogramming, 22(1988)4.
- [9] R.B.Cooper: Introduction to Queuing Theory, The Macmillan Company, 1972.
- [10] T. Czachórski: Zadanle konserwatora - opis za pomoca aproksymacji dyfuzyjnej, Podstawy Sterowania, Tom 14(1984) z.4.
- [11] S.Fukami: Performance Evaluation Method for Interprocessor Interrupts by Message Passing, Microprocessing and Microprogramming, 23(1988).
- [12] S.Hoener, W.Roehder: Efficiency of a Multi - Microprocessor System with Time Shared Busses, Proc. Euromicro Conference Amsterdam, Oct. 1977.
- [13] K.M. Hou, G. Fontenier, M.Senadji: A Simple Bus Analyzer: Parallel Debugging Tool and Real Time Performance Measure Instrument and its Application, Microprocessing and Microprogramming, 19(1987).
- [14] Intel Corporation, Catalogs.
- [15] N.K.Jaiswal: Priority Queues, Academic Press 1968.
- [16] D.Koenig, D.Stoyan: Methoden der Bedienungstheorie Akademie - Verlag, Berlin-DDR 1976. (pol. Metody teorii obsługi masowej, WNT, Warszawa 1979).
- [17] I. Kopocińska: Zagadnienie konserwatora, Roczniki PTM, Seria III: Matematyka Stosowana II (1974).
- [18] I.Koźniewska, M.Włodarczyk: Modele niezawodności i masowej obsługi, PWN, Warszawa 1978.
- [19] J.M.Kurzberg: On the Memory Conflict Problem in Multiprocessor Systems, IEEE Trans. Comp., No.3, March 1974.
- [20] Kwok-Tung Fung, H.C.Torng: On the Analysis of Memory Conflicts and Bus Contentions in a Multiple - Microprocessor System, IEEE Trans. Comp., No.1, Jan. 1979.
- [21] W.Mayberry, G.Efland: Cache Boosts Multiprocessor Performance, Computer Design, No. 1984.
- [22] Proposed Microcomputer Bus Standard (D796 BUS), IEEE Computer Society Subcommittee, Oct. 1980.
- [23] B.Osecky, D.D.Georg, R.J.Bury: The Design of a General - Purpose Multiple Processor System, HP Journal, March 1984.
- [24] J.H.Patel: Analysis of Multiprocessors with Private Cache Memories, IEEE Trans. Comp., No.3, March 1974.
- [25] G.Reyling, Jr: Performance and Control of Multiple Microprocessor Systems, Computer Design, March 1974.

- [26] A.Schmidt: Realizacja systemu wielomikroprocesorowego z buforowanym dostępem do magistrali globalnej, Zeszyty Naukowe Politechniki Śląskiej, Seria Automatyka, Nr 103.
- [27] U.Trottenberg, K.Solchenbach: Parallele Algorithmen und ihre Abbildungen auf parallele Rechnerarchitekturen, Informationstechnik it, 30(1988)2.
- [28] S.Węgrzyn: Podstawy Informatyki, PWN, Warszawa 1982.
- [29] A.Wolisz, T.Czachórski: Systemy obsługi priorytetowej zbioru źródeł jednowymiarowych, Podstawy Sterowania, Tom 3(1973), z.4.
- [30] F.Zitek: Stracony czas. Elementy teorii obsługi masowej, PWN, Warszawa 1974.

Recenzent: Doc.dr hab. Tadeusz Czachórski

Wpłynęło do Redakcji 5.01.1988 r.

ИЗБРАННЫЕ МЕТОДЫ ОПИСАНИЯ ОДНОМАГИСТРАЛЬНОЙ МНОГОПРОЦЕССОРНОЙ СИСТЕМЫ И ИХ ПРАКТИЧЕСКАЯ ПРОВЕРКА

Р е з ю м е

В работе сравнено несколько теоретических методов описания одномагистральной многопроцессорной системы с моделью, учитывающей действительные параметры. Для этого, после определения параметров, описывающих систему, представлены простые формулы, дающие возможность определения максимальной и минимальной эффективности. Далее представлены модели, использующие теорию массового обслуживания, анализ степени использования памяти и вычисляющие полное правдоподобие конфликта. Для многопроцессорной системы проанализирован процесс сотрудничества микропроцессор-магистраль, определяя действительные параметры, учтенные в теоретических моделях. Учитывающая их новая модель сравнена с предыдущими, выясняя причину отличий. Приложены результаты эксперимента, тщательного исследования системы, констатируя соответствие введенной модели с действительностью.

COMPARISON AND EVALUATION OF SOME MODELS OF A SINGLE BUS MULTIPROCESSOR SYSTEMS

Summary

In the paper some theoretical models of a single bus multiprocessor system are compared with one considering real parameters.

First some indices used for description are defined, then simple formulas predicting minimal and maximal performance are given. Next three models of the considered system are presented, the first of them based on the queuing theory, the second on memory utilization comparison and the third on evaluation of a total probability of a conflict. For an existing multimicroprocessor system the transfer between a microprocessor system and a bus was examined and some additional parameters defined. A new model considering these parameters is constructed to be compared with the previous ones. Differences between them are shown and explained. Results of experiments are given to evaluate the model.