

Dariusz BADURA
Włodzimierz WRONA
Instytut transportu
Politechniki Śląskiej

SYMULACJA RUCHU DROGOWEGO W SYSTEMIE WIELOPROCESOROWYM PRZY ZASTOSOWANIU MODELU OPARTEGO O SIEĆ PETRIEGO

Streszczenie. W artykule przedstawiono krótki opis sieci Petriego i sposób jej wykorzystania do budowy symulacyjnego modelu ruchu w sieci ulic i skrzyżowań. Przedstawiono przykładowy model ruchu pojazdu na skrzyżowaniu dla symulacji prowadzonej w systemie wieloprocessorowym oraz alternatywne rozwiązanie wybranego elementu modelu. Rozpatrzono problem rozdziału zadań w systemie symulacyjnym na poszczególne podsystemy z punktu widzenia efektywności pracy całego systemu.

W drugiej części artykułu pokazano, wykorzystanie sieci Petriego do analizy pracy systemu symulacyjnego opartego o mikrokomputerowy system wieloprocessorowy. Ta cecha pozwoliła na uwzględnienie specyfiki stosowanego sprzętu komputerowego na etapie budowy modelu oraz dokonanie wyboru architektury systemu dla realizacji postawionego przed nim zadania. W tym celu zaproponowano budowę makromodelu, tzn. sieci Petriego opisującej najistotniejsze cechy modelu ruchu oraz cechy systemu wieloprocessorowego o proponowanej architekturze.

Znane są dwie grupy modeli ruchu pojazdów:

- modele makroskopowe (np. TRANZYT [1]), ujmujące ruch pojazdów globalnie oraz
- modele mikroskopowe (np. UNITS [2]), rozpatrujące poszczególne zdarzenia dla każdego symulowanego pojazdu.

Dla mikromodeli można przyjąć metodę kolejnych zdarzeń lub metodę stałego kroku.

Tworzenie modelu ruchu miejskiego musi uwzględniać następujące, omówione w licznych publikacjach, zagadnienia:

- stopień dokładności,
- stopień szczegółowości modelu oraz
- liczbę uwzględnionych parametrów.

Niezwykle istotnym problemem pojawiającym się na etapie syntezy modelu jest również uwzględnienie specyfiki maszyny cyfrowej, na której ma lub może zastać model zrealizowany. Muszą przy tym zostać rozpatrzone najnowsze osiągnięcia i możliwości realizacji modelowania za pomocą maszyn cyfrowych; dotyczy to zarówno architektury, jak i oprogramowania [3].

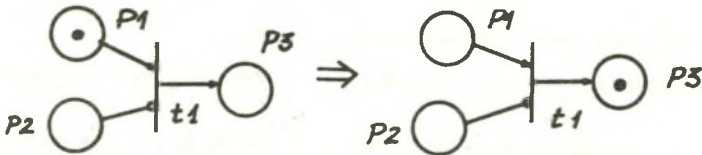
Zadaniami stawianymi symulatorom ruchu drogowego są [1]:

- określenie przepustowości wlotów skrzyżowań, wydzielonych pasów ruchu i innych elementów drogi,
- badanie efektywności różnych systemów organizacji ruchu, m.in. koordynacji sygnalizacji świetlnej i sygnalizacji aktualizowanej, wraz z wyznaczeniem takich parametrów, jak: straty czasu, liczba zatrzymań, długości kolejek pojazdów oraz zużycie paliwa itp.

Jednym z celów stawianych symulatorom ruchu dla zwiększenia jego efektywności jest wspomaganie pracy akomodacyjnego systemu sterowania ruchem w sieci ulic i skrzyżowań. Zatem istotną cechą symulatora powinna być duża szybkość pracy, w niewielkim stopniu zależna od liczby węzłów rozpatrywanej sieci.

Przy dużej liczbie węzłów komunikacyjnych nawet bardzo szybki mikrokomputer nie jest w stanie spełnić tego wymogu. Natomiast może je spełnić system wieloprocesorowy o dobranej architekturze i przy odpowiednim podziale zadań.

W celu przeprowadzenia właściwego rozdziału zadań systemu symulacyjnego na poszczególne procesory niezbędne jest zastosowanie jednolitego aparatu formalnego dla zapisu zdarzeń zarówno w modelu symulacyjnym, jak również w systemie wieloprocesorowym. Zadanie to może spełnić sieć Petriego. Umożliwia ona odwzorowanie własności dynamicznych badanego systemu, a także ich analizę.



Rys. 1. Graficzna reprezentacja wybranego elementu sieci Petriego
Fig. 1. Graphical representation of the chosen element of the Petri net

Graficzną reprezentacją sieci Petriego jest graf zorientowany z dwoma rodzajami wierzchołków zwanych miejscami i przejściami (rys. 1). Łuki skierowane grafu łączą miejsca wejściowe (P_1 , P_2) z przejściami (t_1) i przejścia z miejscami wyjściowymi (P_3). Rozróżniamy dwa rodzaje łuków: zakończone grotem i zakończone kółkiem. Każde miejsce może zawierać jeden lub większą liczbę markerów reprezentowanych w sieci przez kropki. Przejście jest

aktywne, gdy wszystkie miejsca połączone z nim łukami z grotem zawierają markery, a miejsca połączone łukami z kółkiem nie zawierają markerów. Zapalenie przejścia polega na usunięciu markera z każdego miejsca wejściowego połączonego z przejściem łukiem zakończonym grotem, a dodanie markera do każdego miejsca wyjściowego. Tranzycje w sieci Petriego (przejścia) określają występujące zdarzenia, natomiast miejsca przedstawiają warunki występowania zdarzeń [4].

Przedmiotem artykułu jest wykorzystanie sieci Petriego do budowy modelu symulacyjnego i analizy możliwości realizacji modelu w oparciu o wieloprocesorowy system mikrokomputerowy. W tym celu przedstawiono:

- przykładowy, oparty na sieci Petriego, model ruchu pojazdów na skrzyżowaniu z sygnalizacją świetlną i w połączeniu między skrzyżowaniami,
- sposób podziału zadań w systemie symulacyjnym,
- zastosowanie sieci Petriego do analizy pracy systemu symulacyjnego w układzie wieloprocesorowym oraz
- próbę wyboru struktury układowej i programowej systemu wieloprocesorowego do celów symulacji.

O zastosowaniu sieci Petriego do budowy modelu ruchu drogowego zadecydowały:

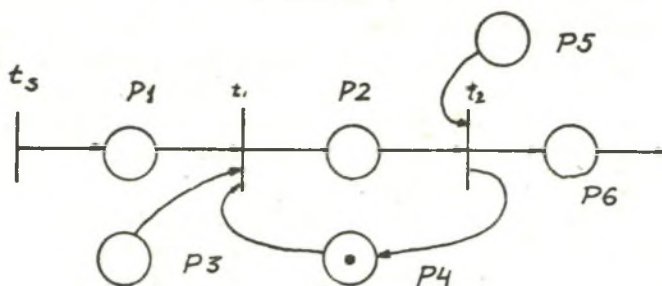
- uniwersalność zastosowania aparatu formalnego zarówno do tworzenia modelu, jak i analizy pracy symulatora opartego na systemie wieloprocesorowym,
- przejrzystość tworzenia modelu ruchu pojazdów,
- możliwość tworzenia modeli nietypowych węzłów komunikacyjnych (np. rondo),
- uproszczenie rozbudowy programowej i urządzeniowej systemu.

Budowa modelu symulacyjnego ruchu drogowego

Model symulacyjny ruchu drogowego sieci ulic i skrzyżowań z sygnalizacją świetlną utworzony za pomocą sieci Petriego można przedstawić jako połączenie kilku modułów charakteryzujących zachowanie się pojazdu i symulujących jego ruch. W zależności od dokładności przeprowadzonych badań każdy element modelu można dowolnie uszczegóławiać i modyfikować zachowując kompatybilność.

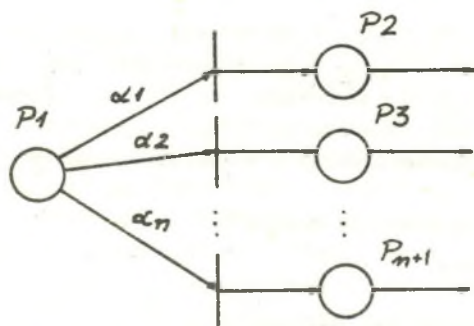
Poniżej rozpatrywane będą zasadnicze elementy modelu:

- przejazd samochodu przez skrzyżowanie,
- wybór kierunku lub pasa ruchu,
- mechanizm odzwierciedlający upływ czasu rzeczywistego,
- przejazd pojazdu z jednego skrzyżowania do drugiego oraz
- sytuacje kolizyjne.



Rys. 2. Graficzny model przejazdu pojazdu przez skrzyżowanie
 Fig. 2. Graphical model of the movement of the vehicle through the cross-road

Przejazd samochodu przez skrzyżowanie w modelowanej strukturze sieci przedstawia rys.2. Pojazd generowany jest według danego przyjętego rozkładu prawdopodobieństwa w zależności od organizacji ruchu i natężeń pojazdów na wybranym skrzyżowaniu [5]. Na rys. 2 źródło t_s połączone strzałką z miejscem P1 reprezentuje generator pojazdów. Generowany pojazd wprowadzany jest do kolejki przed skrzyżowaniem. W przypadku wystąpienia zielonego światła na danym kierunku (marker w miejscu P3) i wolnego miejsca na skrzyżowaniu (marker w miejscu P4) pojazd przejeżdża przez linię stop, tzn. że spełnione zostały warunki do wystąpienia przejścia t_1 i pojawia się na skrzyżowaniu (marker w miejscu P2). Pojazd znajdujący się na skrzyżowaniu może je opuścić - zapalenie miejsca t_2 - tylko w przypadku spełnionego warunku reprezentowanego na rys. 2 w postaci miejsca P5 (marker w miejscu P5). Pojazd opuszczający skrzyżowanie wjeżdża na jeden z pasów ruchu za skrzyżowaniem. Na rys. 2 w miejscu P6 pojawia się marker.

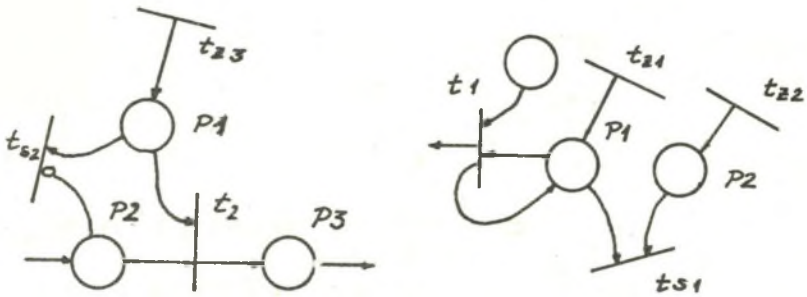


Rys. 3. Konflikt w sieci Petriego
 Fig. 3. Conflict in Petri net

Analiza sieci Petriego pozwala wyróżnić zjawiska wymagające przyjęcia dodatkowych zasad tworzenia sieci. Jednym z takich zjawisk jest występowanie konfliktu. Przykładem tego w omawianym modelu symulacyjnym jest wybór kierunku lub pasa ruchu (rys. 3). Zjawisko to rozwiązywane jest poprzez określenie prawdopodobieństwa wystąpienia rozpatrywanego zdarzenia. Po wygenerowaniu pojazdu (marker w miejscu P1) następuje wybór przejścia, które wystąpi.

Na rys. 3 prawdopodobieństwo wystąpienia danego zdarzenia określają współczynniki α przypisane każdemu przejściu. Suma prawdopodobieństwa dla odrębnego jednego zespołu przejść tworzącego konflikt równa się jeden. Uniwersalność modelu narzuca przyjęcie maksymalnej liczby pasów, jaka może w rzeczywistości wystąpić przed skrzyżowaniem. Rozwiązanie konkretnego zadania przy mniejszej ilości pasów sprowadza się do przypisania nieistniejącym pasom lub kierunkom ruchu prawdopodobieństwa równego zero.

Składnikiem każdej metody symulacyjnej jest mechanizm odzwierciedlający upływ czasu rzeczywistego. Wybrany model symulacyjny opiera się na metodzie kolejnych zdarzeń [6]. W przyjętym modelu określenie czasu stanowi podstawowy warunek zastosowania sieci Petriego do symulacji. Upływ czasu nie jest procesem ciągłym, lecz dyskretnym. Podaje się początek i koniec czasu trwania wybranego procesu symulacji lub odstępy czasu pomiędzy procesami.



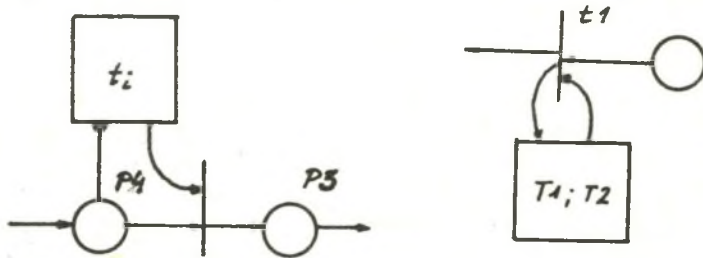
Rys. 4. Mechanizm odzwierciedlający upływ czasu w sieci Petriego
a) przez podanie momentów początku i końca, b) przez podanie długości odstępu pomiędzy zdarzeniami

Fig. 4. Mechanism illustrating passage of time in the Petri net
a) by giving the beginning and the end, b) by giving time intervals between events

W sieci Petriego wykorzystywanej w modelu symulacyjnym realizuje się to przez:

- układ przedstawiony na rys. 4a, gdzie początek procesu jest wystąpieniem tranzycji t_{21} (np. zapalenie światła czerwonego na danym kierunku ruchu), a koniec to wystąpienie tranzycji t_{22} (np. zapalenie światła zielonego),
- układ przedstawiony na rys. 4b, gdzie odstęp czasu pomiędzy symulowanymi procesami (np. przejazd pojazdu przez linię stop) regulowany jest wystąpieniem tranzycji t_{23} .

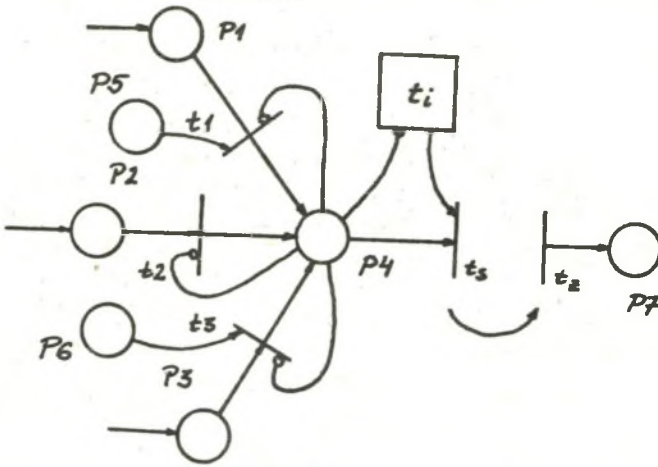
W układzie pierwszym (rys. 4a) zapalenie przejścia t_{z1} wpisuje marker do miejsca P1. Pojawienie się markera w miejscu P1 umożliwia dowolne zapalenie się przejścia t_1 w zależności tylko od warunków sieci niezależnych od czasu. Po zapaleniu przejścia t_{z2} pojawia się marker w miejscu P2, co powoduje zapalenie przejścia t_{s1} i usunięcie markerów z miejsc P1 i P2 i zablokowanie przejścia t_1 . W układzie drugim (rys. 4b) zapalenie przejścia t_{z3} wpisuje marker do miejsca P3, co umożliwia wystąpienie tranzycji t_2 , pod warunkiem że w miejscu P4 oczekuje marker (np. pojazd). W przypadku braku markera w miejscu P4 wystąpi tranzycja t_{s2} , która usuwa marker z miejsca P1. Proces powtarza się w odstępach czasu określonych wybranym rozkładem prawdopodobieństwa. Układ z rys. 4a będziemy zastępowali symbolem przedstawionym na rys. 5a, a układ z rys. 4b symbolem z rys. 5b.



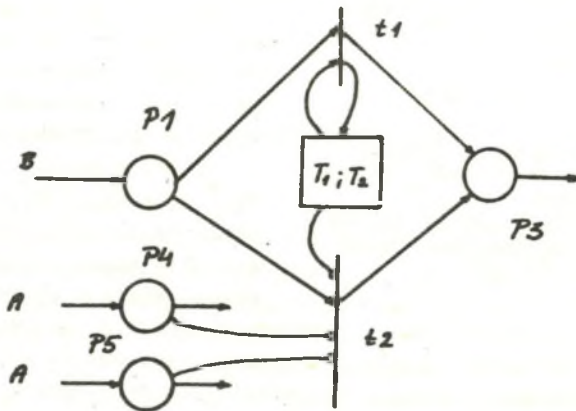
Rys. 5. Symbol reprezentujący układ z rysunku
a) 4a, b) 4b

Fig. 5. Symbols representing a system in fig.
a) 4a, b) 4b

Istotnym problemem symulacji ruchu drogowego w sieci ulic i skrzyżowań jest przejazd pojazdu z jednego skrzyżowania na wlot drugiego, a także sumowanie się pojazdów na pasie wylotowym z różnych kierunków ruchu. W proponowanym modelu przedstawia to rys.6, który również zawiera błądzek czasowy z rys. 5b, odzwierciedlający przejazd pojazdów z pewnym odstępem czasowym t_1 . Pojazdy znajdujące się na skrzyżowaniu mogą ze wszystkich możliwych kierunków ruchu (miejsca P1, P2, P3) wyjechać na jeden z dowolnych wylotowych pasów ruchu (np. miejsce P4), pod warunkiem że miejsce na tym pasie jest wolne, tj. każda z tranzycji t_1 , t_2 , t_3 blokowana jest zajętością miejsca P4. Opuszczenie miejsca wylotowego na danym pasie ruchu związane jest z wystąpieniem tranzycji t_s , która powoduje wygenerowanie pojazdu t_z i pojawienie się go w kolejce (P7) na następnym skrzyżowaniu po czasie, jaki wymagany jest na przejazd tego pojazdu między skrzyżowaniami (offset). Pojazdy opuszczające skrzyżowanie danym pasem ruchu wjeżdżają na to skrzyżowanie z dowolnych kierunków tworząc między sobą kolizje. Na rys. 6 przedstawiają to miejsca P5 i P6.



Rys. 6. Graficzny model przejazdu pojazdu pomiędzy skrzyżowaniami
 Fig. 6. Graphical model of the movement of the vehicle between cross-roads



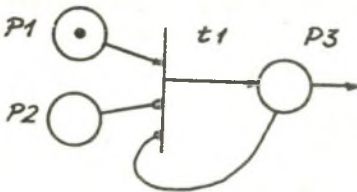
Rys. 7. Przykład kolizji w modelu ruchu drogowego
 Fig. 7. An example of a collision in the traffic model

Podobna kolizja (rys. 7) występuje pomiędzy pojazdami skręcającymi w lewo (kierunek B) i jadącymi z przeciwnej strony na wprost (kierunek A). W przypadku kolizji w czasie ruchu pojazdów muszą być zachowane przepisy ruchu drogowego. Uwzględniając tę zasadę tworzy się w modelu symulacyjnym priorytetowe kierunki warunkujące poruszanie się pojazdów znajdujących się na kierunkach podrzędnych. Ruch pojazdów ulegających kolizji jest możliwy w następujących przypadkach:

- przy wystąpieniu czerwonego światła na kierunkach nadrzędnych (tranzycja t_1),
- przy braku pojazdów jadących z kierunków nadrzędnych i mających zielone światło (tranzycja t_2).

Na rys. 7 czas T_1 określa moment zapalenia się światła czerwonego dla kierunku A, natomiast czas T_2 określa moment zapalenia się światła zielonego dla tego kierunku. Kolidacja sygnalizowana jest zajętością miejsc P4, P5 z kierunku A. Miejsce P1 oznacza miejsce oczekiwania pojazdu przed opuszczeniem skrzyżowania, a miejsce P3 oznacza miejsce na pasie wylotowym ze skrzyżowania.

Powyższy sposób modelowania wybranych elementów ruchu drogowego w sieci ulic i skrzyżowań nie wyczerpuje wszystkich możliwych rozwiązań tego zagadnienia [3]. Sposób ten jest arbitralnie przyjętym zapisem ruchu wygodnym przy projektowaniu i określaniu parametrów ruchu w przyjętej sieci.



Rys. 8. Graficzny model przejazdu pojazdu przez skrzyżowanie (wariant II)

Fig. 8. Graphical model of the vehicle movement through the crossroad (variant II)

przejściem t_1 za pomocą łuku z kółkiem. Powyższe dwa rozwiązania modelowe (rys. 2 i 8) przejazdu pojazdu przez linię stop różnią się jedynie układem miejsc i przejść w modelu, natomiast warunki i sposób poruszania się pojazdu są identyczne.

Model oparty na sieci Petriego umożliwia symulację ruchu drogowego na dowolnym komputerze, ale z powodów, o których już wspomniano we wstępie, w dalszej części artykułu omówiony zostanie proces symulacji prowadzony w systemie wieloprocesorowym, którego struktura programowa i układowa przystosowana jest do symulacji ruchu.

Jako przykład tworzenia różnych konfiguracji modelowych tych samych zagadnień w modelu symulacyjnym można przedstawić element opisujący przejazd pojazdu przez linię stop (rys. 8). Element ten był już wcześniej prezentowany na rys. 2. Miejsce P1 przedstawia obecność pojazdu na linii stop, miejsce P2 - palenie się czerwonego światła na wlocie, miejsce P3 - obecność pojazdu na skrzyżowaniu, zaś przejście t_1 - wjazd pojazdu na skrzyżowanie. Miejsce P3 jest połączone z

Architektura systemu wieloprocesorowego do celów symulacji ruchu

System wieloprocesorowy wykorzystywany do procesu symulacji powinien spełniać wymagania co do szybkości i niezawodności pracy poszczególnych modułów, a także powinien posiadać odpowiednią dostosowaną do problemu symulacyjnego pamięć systemową. Szybkość pracy systemu nie tylko opiera się na możliwościach poszczególnych procesorów, ale również zależy w dużym stopniu od jego struktury układowej i programowej. System ten musi zapewnić pełną realizowalność postawionego zadania, a także powinien mieć możliwość rozbudowy bez naruszania struktury całego układu. Prowadzi to do rozważań nad architekturą tego systemu wieloprocesorowego i jego programu pracy, spełniającego wyżej wymienione założenia. Już w pierwszym etapie projektowania systemu (określenia schematu blokowego) nasuwa się pytanie, jaką strukturę systemu przyjąć: czy przyporządkować każdemu węzłowi komunikacyjnemu jeden procesor z pamięcią lokalną, czy też połączyć procesory jednego ciągu skrzyżowań wspólną magistralą ze wspólną pamięcią.

Proponowana struktura systemu wieloprocesorowego służącego do celów symulacji powinna zawierać następujące elementy:

- magistralę systemową wraz ze wspólną pamięcią,
- mikroprocesory z arbitrami magistrali oraz magistralami lokalnymi, pamięciami, układami wejścia-wyjścia lub kanałami bezpośredniego dostępu do pamięci DMA,
- układ arbitrażu,
- podsystem mikrokomputerowy-podporządkowany z układem V24 i rejestrem liniowym [7].

Magistrala systemowa umożliwia bezpośredni dostęp do pamięci RAM dla każdego mikroprocesora systemu zawierającego arbiter. W pamięci wspólnej RAM przechowywane są tablice obejmujące dane o pojazdach znajdujących się na połączeniu między dwoma skrzyżowaniami oraz inne dane będące przedmiotem wymiany między mikroprocesorami.

Mikroprocesory magistrali systemowej przeznaczone są do sterowania procesami symulacji ruchu ulicznego na jednym lub kilku skrzyżowaniach. Program i dane symulacji przekazywane są do pamięci lokalnej procesorów i tam przechowywane. Pamięć lokalna dołączona jest do magistrali lokalnej, do której dołączone są również układy wejścia-wyjścia i układy pośredniczące. Układy we-wy służą do komunikacji mikroprocesora z systemem mikrokomputerowym-podporządkowanym lub do realizacji sprzętowej zegara systemowego. Układy pośredniczące (arbiter) służą do połączenia podsystemów (mikroprocesorów) z magistralą systemową.

Arbiter magistrali umożliwia (realizuje) połączenie danego procesora z pamięcią wspólną systemu, zaś układ arbitrażu wyznacza, który z procesorów może zostać aktualnie dołączony do magistrali systemowej. Układ arbitrażu może uwzględniać zależności priorytetowe lub może być jednostką posiadającą dodatkowy procesor, który testując parametry wszystkich podsystemów żądają-

cych połączenia z pamięcią wspólną adaptacyjnie przydziela magistralę systemową wybranemu podsystemowi.

Podsystem mikrokomputerowy-podporządkowany pełni rolę służebną wobec całego systemu:

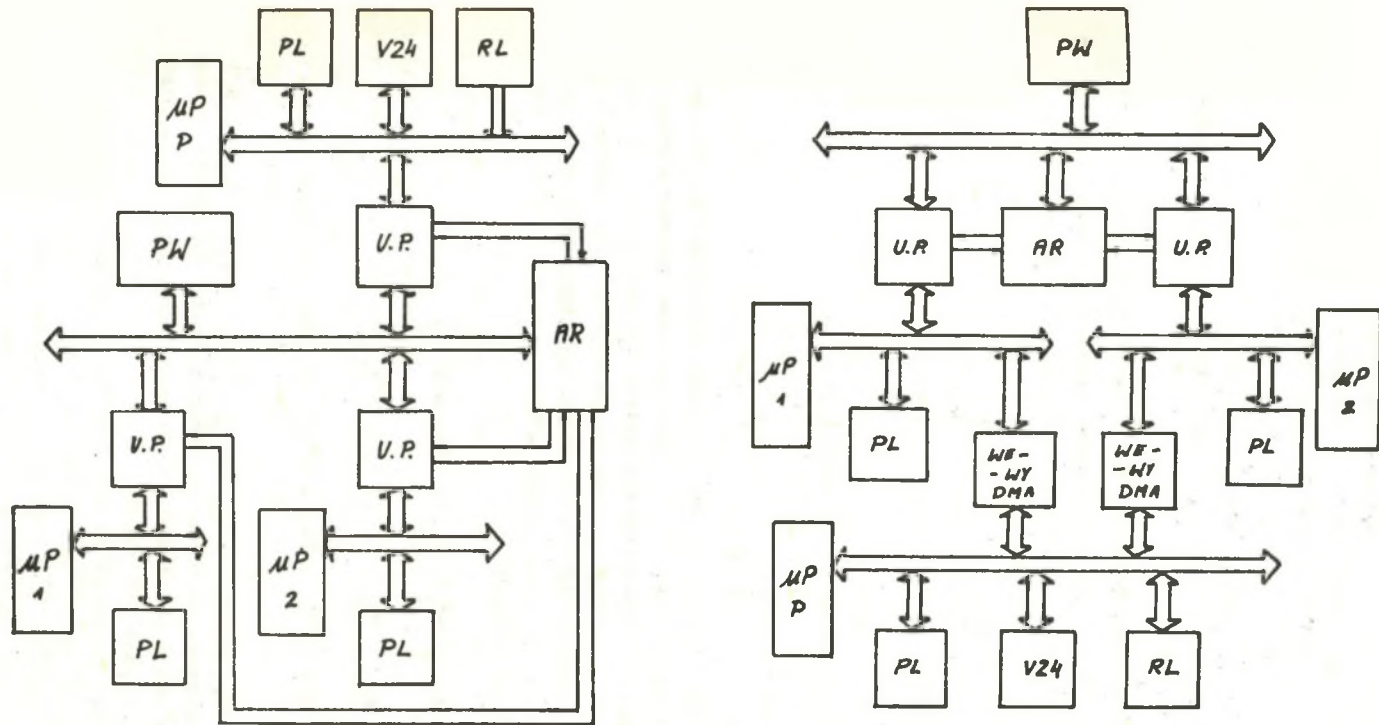
- generuje liczby losowe dla potrzeb poszczególnych symulatorów,
- umożliwia dostęp danych dla użytkownika,
- wprowadza dane wejściowe,
- opracowuje i wyprowadza dane wyjściowe.

Generacja liczb losowych odbywa się sprzętowo poprzez dołączenie do magistrali mikrokomputera-podporządkowanego rejestru liniowego z własnym generatorem lub z generatorem systemowym (generator impulsów synchronizujących system mikrokomputerowy-podrzędny). W pierwszym przypadku używa się dużą niezależność generowanych liczb, zaś w drugim wynik symulacji jest powtarzalny. Kontakt z operatorem umożliwia podłączony do interfejsu V24 monitor bądź drukarka z klawiaturą.

Wybrane przykładowe dwie struktury układowe systemu wieloprocesorowego oparte na powyższych zasadach przedstawione są na rys. 9a i 9b. Rys. 9a przedstawia system ze wspólną pamięcią, do której mają dostęp wszystkie podsystemy systemu łącznie z podsystemem mikrokomputerowym-podporządkowanym. Wspólna pamięć pozwala na wymianę informacji pomiędzy niezależnymi procesorami (np. wymiana pojazdów). Systemy wieloprocesorowe są najczęściej reprezentowane przez taką strukturę. Przykładem takiej struktury jest system ELWRO 800 opracowany przez Instytut Automatyki Politechniki Poznańskiej i Instytut Systemów Automatyki we Wrocławiu. Jest to rodzina komputerów oparta na mikroprocesorach 8- i 16-bitowych [8].

Struktura druga przedstawiona jest na rys. 9b. System mikrokomputerowy-podporządkowany komunikuje się z poszczególnymi podsystemami poprzez układy we-wy lub kanał DMA bezpośrednio z poszczególnymi pamięciami tych procesorów. W strukturze tej występuje również magistrala systemowa ze wspólną pamięcią.

Przyjęcie pewnej konfiguracji systemu wieloprocesorowego nie określa ostatecznie programu pracy tego systemu. Przykładem może być przesłanie pojazdu z jednego układu (skrzyżowania) do drugiego, co wymaga uruchomienia układu arbitrażu. Układ ten dopuszcza systemy do wspólnej pamięci systemowej w celu wpisania lub odczytania pojazdów wjeżdżających lub wyjeżdżających ze skrzyżowania. Proces wymiany liczby pojazdów między skrzyżowaniami określony jest przyjętym programem obsługi arbitrażu. Podsystem może zgłaszać zapotrzebowanie na komunikację z pamięcią albo przy każdym opuszczającym skrzyżowanie pojeździe, albo może gromadzić pojazdy wyjeżdżające ze skrzyżowania w pamięci lokalnej, a następnie większą ich partię wpisywać do pamięci wspólnej.



Rys. 9. Schemat blokowy struktury układowej systemu wieloprocessorowego

a) wszystkie procesory podłączone do wspólnej magistrali, b) procesor podrzędny połączony z podsystemami za pomocą układów we-wy lub kanału DMA

Fig. 9. Block diagram of the system structure of the multiprocessor system

a) all processors connected to the common bus, b) subprocessor connected with subsystems by input-output systems or the DMA

Istotnym zagadnieniem, które pojawia się przy współpracy podsystemów, są zależności czasowe pomiędzy zegarami systemowymi podsystemów. Decydują one o efektywności pracy całego systemu i przebiegu procesu symulacji. Oznaczmy przez:

TS_i - stan zegara systemowego i -tego podsystemu,

T_{ij} - minimalny czas przemieszczania elementu (pojazdu) z i -tego do j -tego podsystemu oraz

Δt_i - wyznaczony na podstawie rozpatrywanych zdarzeń przyrost stanu TS_i .

Wtedy dla dwóch odizolowanych podsystemów otrzymujemy następujące zależności opisujące prawidłową (bez zatrzymań) pracę całego systemu:

$$\left. \begin{aligned} TS_i + \Delta t_i &\leq TS_j + T_{ji} \\ TS_j + \Delta t_j &\leq TS_i + T_{ij} \end{aligned} \right\} \text{ lub } \quad (1)$$

Stąd:

$$\left. \begin{aligned} TS_i &\leq TS_j + (T_{ji} - \Delta t_i) \\ TS_j &\leq TS_i + (T_{ij} - \Delta t_j) \end{aligned} \right\} \text{ lub } \quad (2)$$

Gdy $(T_{ji} - \Delta t_i)$ oraz $(T_{ij} - \Delta t_j)$ są dodatnie, tzn. częstość zdarzeń w podsystemie jest większa od odwrotności czasu przemieszczania elementów pomiędzy podsystemami, to zawsze system będzie czynny, a więc co najmniej jeden podsystem będzie czynny. W przeciwnym wypadku może nastąpić zatrzymanie systemu lub "zgubienie" niektórych zdarzeń spowodowane niespełnieniem warunku (1) i (2). Dla trzech podsystemów i, j, k otrzymamy następujące zależności czasowe:

$$\left. \begin{aligned} TS_i + \Delta t_i &\leq \text{Min} \left\{ (TS_j + T_{ji}), (TS_k + T_{ki}) \right\} \\ TS_j + \Delta t_j &\leq \text{Min} \left\{ (TS_i + T_{ij}), (TS_k + T_{kj}) \right\} \\ TS_k + \Delta t_k &\leq \text{Min} \left\{ (TS_i + T_{ik}), (TS_j + T_{jk}) \right\} \end{aligned} \right\} \quad (3)$$

Gdy $(T_{ji} - \Delta t_i)$, $(T_{ki} - \Delta t_i)$, $(T_{ij} - \Delta t_j)$, $(T_{kj} - \Delta t_j)$, $(T_{ik} - \Delta t_k)$ oraz $(T_{jk} - \Delta t_k)$ są dodatnie, co najmniej jeden mikrosystem będzie czynny (podobnie jak dla dwóch podsystemów). Podobne rozważania można przeprowadzić dla większej liczby podsystemów.

Powyższe nierówności prowadzą do wniosku, że dla efektywnej pracy systemu wieloprocesorowego podział zadań powinien dążyć do maksymalizacji czasów przemieszczania elementów pomiędzy mikrosystemami przy zachowaniu odpowiednio dużej częstości zdarzeń w poszczególnych podsystemach.

Częstotliwość uaktualniania przez poszczególne mikrosystemy informacji o czasach systemowych sąsiednich podsystemów wpływa na ograniczenie realnych czasów systemowych i może spowodować nieuzasadnione "spowolnienie" pracy systemu. Na częstotliwość ma wpływ zarówno sposób programowej obsługi informacji o czasach systemowych poszczególnych podsystemów, jak i architektura całego systemu. Możliwe jest np. umieszczenie czasu systemowego w rejestrze, do którego mają dostęp sąsiednie procesory, bezpośrednio przez układy we-wy.

Podział zadań w systemie powinien zmierzać również do maksymalnie wyrównanej prędkości przyrostu czasów systemowych, np. jeden mikrosystem może symulować odpowiednią liczbę skrzyżowań, mimo iż z punktu widzenia prostoty oprogramowania jednemu procesorowi należałoby przyporządkować jedno skrzyżowanie.

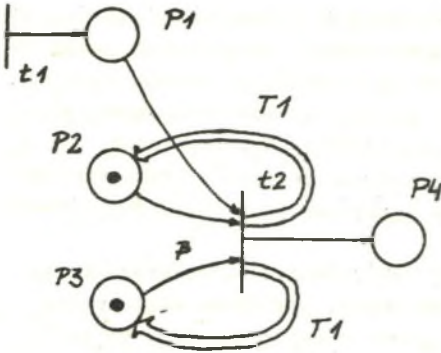
Struktura układowa i programowa systemu wieloprocesorowego służącego do celów symulacji ruchu drogowego zawiera między innymi elementy związane tylko z procesem symulacyjnym. Konfiguracja układowa systemu powinna posiadać podsystem mikrokomputerowy-podporządkowany wyznaczający liczby losowe oraz umożliwiający dostęp danych do poszczególnych symulatorów. W programie pracy systemu powinien wystąpić podprogram testowania zegara systemowego prowadzący do maksymalizacji czasów przemieszczania elementów pomiędzy podsystemami symulacyjnymi. Dla znalezienia optymalnej struktury układowej i programowej systemu konieczne jest określenie metody wyboru optymalnego rozwiązania.

Wybór optymalnej struktury systemu wieloprocesorowego

Z powyższych rozważań wynika, że istotne staje się stworzenie aparatu pozwalającego w prosty sposób dokonać wyboru struktury układowej i programowej systemu wieloprocesorowego do celów symulacji. Wybór optymalnej struktury systemu zależy od przyjętych kryteriów jakości oraz od parametrów symulowanej sieci ulic i skrzyżowań. Wydaje się słuszne przyjęcie następujących kryteriów jakości dla projektowanego systemu:

- minimalny czas symulacji ruchu pojazdów,
- prostota konstrukcji i łatwość programowania systemu wieloprocesorowego oraz
- stopień niezależności prawdopodobieństwa występowania zdarzeń.

Analiza poszczególnych wariantów systemu wieloprocesorowego wymaga utworzenia modelu tego systemu. Jak już we wstępie podano, w dalszej części artykułu zostanie przedstawiony jeden z możliwych modeli opisujących system wieloprocesorowy służący do celów symulacji. Model ten oparty jest również na sieci Petriego. Przedstawia działanie systemu przyjmując zasadę sprowadzenia przeprowadzanych w systemie operacji do czasów ich trwania oraz sprawdzania warunków występowania zdarzeń procesu symulacji.



Rys. 10. Przedstawienie czasu operacji w modelu systemu wieloprocesorowego

Fig. 10. Representation of time of the operation in the microprocessor system model

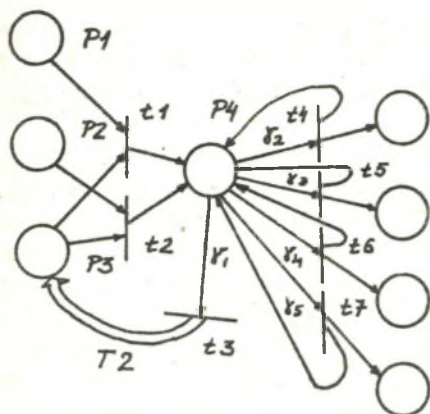
W prezentowanym modelu czas trwania poszczególnych operacji symbolicznie oznacza się podwójną strzałką tylko od przejścia do miejsca. Na rys. 10 prezentowane jest oznaczenie czasu $T1$ operacji zapisu informacji (np. wjazd pojazdu do kolejki na skrzyżowaniu) do pamięci procesora. Po zapaleniu przejścia $t1$ w miejscu $P1$ pojawia się pojazd. Jeżeli wolny jest w tym czasie procesor, a także wolna jest pamięć (marker w miejscu $P2$ i $P3$), pojazd znajdujący się w miejscu $P1$ może zostać wpisany do pamięci w miejsce $P4$. Wykona-

nie zapisu informacji do pamięci przez procesor zależy również od tego, czy procesor nie rozpocznie wykonywania innej operacji o wyższym priorytecie. W modelu priorytet wykonywania poszczególnych operacji oznacza się współczynnikiem β . Wpisanie pojazdu do pamięci reprezentowane jest na rys. 10 przez zapalenie przejścia $t2$.

Operacja zapisu do pamięci trwa przez pewien stały okres czasu $T1$ i zależy od szybkości procesora, a także od podprogramu realizującego tę operację. Podobnie oznacza się w modelu czas trwania poszczególnych operacji złożonych z jednej lub wielu instrukcji.

Model złożony jest z podmodeli opisujących działanie poszczególnych podsystemów połączonych modelem arbitra i pamięci wspólnej w przypadku struktury pierwszej, a także układem we-wy lub kanałem DMA dla drugiej struktury. Procesor, pamięć, układ arbitrażu reprezentowane są w modelu przez miejsca. Obecność markera w miejscu oznacza stan spoczynku tych układów, zaś brak markera zajętość tych układów.

Wybrany element modelu systemu wieloprocesorowego opisujący proces testowania sieci Petriego reprezentującej model ruchu drogowego w tym systemie przedstawia rys. 11. Testowanie sieci Petriego polega na tym, że po wystąpieniu w procesie symulacji zdarzenia sprawdza się warunki sieci aż do momentu, kiedy wszystkie z nich są niespełnione. Zdarzenie następne może wystąpić dopiero po modyfikacji zegara systemowego do wartości reprezentującej moment wystąpienia tego zdarzenia. Na rys. 11 miejsce $P1$ określa pojawienie się przed kolejką na skrzyżowaniu pojazdów z innych skrzyżowań, miejsce $P2$ przedstawia inne zdarzenia modelu ruchu drogowego (np. moment zapalenia światła zielonego dla danego kierunku ruchu). Jeżeli procesor jest wolny i jeżeli kolejne zdarzenie oczekuje na realizację (marker w miejscu $P1$ lub $P2$), to zachodzi zdarzenie $t1$ lub



Rys. 11. Graficzna reprezentacja wybranego elementu modelu systemu wieloprocesorowego

Fig. 11. Graphical representation of the chosen element of the microprocessor system model

t_2 testowania sieci Petriego modelu symulacyjnego ruchu pojazdów. W modelu zapalenie przejść t_1 i t_2 wprowadza marker do miejsca P_4 . Pojawienie się markera w miejscu P_4 umożliwia zapalenie się jednego z przejść t_3 , t_4 , t_5 , t_6 , t_7 .

W czasie testowania sieci Petriego można wygenerować pojazd do innego skrzyżowania dla dowolnego kierunku ruchu (zdarzenia t_4 , t_5 , t_6 , t_7). Zdarzenie t_3 określa sprawdzenie warunków sieci bez wygenerowania pojazdów do innego skrzyżowania.

Zajścia zdarzeń $t_3 \div t_7$ zależy od współczynników określających rozkład występowania zdarzeń: bądź wyjazdu pojazdów ze skrzyżowania, bądź sprawdzenia warunków sieci bez generacji

pojazdów do innych skrzyżowań. Po sprawdzeniu warunków sieci Petriego (zapalenie przejścia t_3) procesor może rozpocząć wykonywanie następnej operacji. Przejście t_3 po czasie T_2 wpisuje marker w miejsce P_3 reprezentujące stan procesora.

W prezentowanym przykładzie modelu testowania sieci Petriego występują pewne nieokreślone parametry. Są nimi:

- czas T_2 sprawdzania wszystkich warunków sieci pomiędzy występującymi zdarzeniami,
- współczynniki λ , wyznaczające które przejście ($t_3 \div t_7$) i z jakim prawdopodobieństwem zapali się.

W pełnym modelu systemu wieloprocesorowego do celów symulacji takich nieokreślonych parametrów jest więcej. Poza wspomnianymi wyżej czasem T_2 i współczynnikami λ występują jeszcze nieokreślone parametry, takie jak:

- czas zgłoszenia przez procesor zapotrzebowania do arbitra przy opuszczaniu pojazdu skrzyżowania obsługiwanego przez dany procesor,
- czas wpisu do pamięci wspólnej pojazdów z pamięci lokalnej,
- czas testowania stanu komórek pamięci (czy pojazdy oczekują na przepisanie z jednego skrzyżowania na wlot do drugiego?),
- czas podłączenia procesora do pamięci wspólnej po otrzymaniu zezwolenia od arbitra,

- współczynniki określające pierwszeństwo obsługi procesora przez układ arbitrażu itp.

Przedstawione powyżej parametry nieokreślone nie występują w każdej proponowanej strukturze układowej i programowej, są związane z pierwszą przedstawioną już strukturą systemu. Każda zaproponowana struktura posiada własne parametry nieokreślone; mogą w niej pojawiać się nowe parametry typowe tylko dla tej struktury.

Dalszym etapem pracy nad modelem systemu jest znalezienie funkcji lub stałych opisujących wyżej wymienione parametry nieokreślone. Nasuwa się rozwiązanie, aby w procesie symulacji ruchu drogowego na jednym skrzyżowaniu, przeprowadzonej przy wykorzystaniu maszyny cyfrowej, określić pewne wybrane parametry modelu systemu. Proces symulacji pozwoli bezpośrednio wyznaczyć tylko niektóre parametry (np. współczynniki k wyznaczające kolejność zapalenia się przejść $t_3 + t_7$), natomiast pośrednio umożliwi znalezienie czasów wykonywania poszczególnych operacji. Pośrednie wyznaczenie czasów operacji polega na porównaniu sumy czasów wykonywanych instrukcji tej operacji przez symulujący ruch procesor w systemie wieloprocessorowym (suma instrukcji pomnożona przez ilość cykli maszynowych procesora) z minimalnym czasem operacji wyznaczonym w procesie symulacji na m.c. Porównanie tych czasów wyznacza współczynnik proporcjonalności pomiędzy nimi.

Pozostałe czasy trwania poszczególnych operacji wyznacza się sumując czasy wszystkich instrukcji występujących w danej operacji. Przykładem takiego określenia parametrów jest operacja wpisu pojazdu do pamięci wspólnej z pamięci lokalnej. Operacja ta składa się z instrukcji dla mikroprocesora Intel 8080:

MOV A, M (2 cykle maszynowe, 7 taktów zegarowych),

MOV M, A (2 cykle maszynowe, 7 taktów zegarowych).

Instrukcja pierwsza pobrania danej z pamięci lokalnej o adresie znajdującym się w rejestrach H i L i wpisania jej do akumulatora trwa przez $7 \times 2 \mu s$, = $14 \mu s$, natomiast wykonanie instrukcji drugiej wpisującej zawartość akumulatora do pamięci wspólnej o adresie w rejestrach H i L (zawartość rejestrów zmieniona) trwa przez $7 \times 2 \mu s$ = $14 \mu s$. Łączny czas operacji wynosi $28 \mu s$. Powyższy przykład wskazuje, że wartości parametrów nieokreślonych zależą również od przyjętego typu procesora i programu realizującego dane operacje.

Mając model systemu wieloprocessorowego, rozsądne byłoby wybrać również metodę symulacji na m.c. jako metodę wyboru struktury układowej i programowej tego systemu. Metoda symulacji umożliwiła porównanie systemów o różnych strukturach oraz określenie czasów pracy tych systemów.

Definiowanie struktur, przygotowywanie ich i wprowadzanie do programu symulacyjnego przeprowadza projektant systemu. Powstające struktury muszą spełniać założenie poprawnej realizacji postawionego zadania, a także uwzględniać rzeczywisty podział zadań w systemie [9].

Symulacja sytemu wieloprocesorowego, a także symulacja ruchu drogowego dla jednego skrzyżowania odbywa się na m.c. MERA 400 w programie Portran CROOK [10].

Podsumowanie

W artykule pokazano sposób wykorzystania sieci Petriego do budowy symulacyjnego modelu ruchu w sieci ulic i skrzyżowań. Przedstawiono przykładowy model dla symulacji prowadzonej w systemie wieloprocesorowym oraz alternatywne rozwiązanie wybranego elementu modelu. Rozpatrzono problem rozdziału zadań w systemie symulacyjnym na poszczególne podsystemy z punktu widzenia efektywności pracy całego systemu.

W drugiej części artykułu pokazano, iż sieć Petriego może być również wykorzystana do analizy pracy systemu symulacyjnego opartego na mikrokomputerowym systemie wieloprocesorowym. Ta cecha pozwoliła na uwzględnienie na etapie budowy modelu specyfiki stosowanego sprzętu komputerowego oraz dokonanie wyboru architektury systemu dla realizacji postawionego przed nim zadania. W tym celu zaproponowano następujący algorytm postępowania:

- 1) budowa makromodelu systemu, tzn. sieci Petriego opisującej najistotniejsze cechy modelu ruchu oraz cechy systemu wieloprocesorowego o proponowanej architekturze,
- 2) określenie parametrów makromodelu oraz
- 3) przeprowadzenie symulacji utworzonej sieci na m.c.

Przedstawienie działania rozpatrywanego układu za pomocą sieci Petriego umożliwia często prostszą realizację programową układu. Taka reprezentacja sieci stanów i zdarzeń w układzie ułatwia dokonywanie niezbędnej dekompozycji złożonej sieci proponowanej realizacji modelu.

Łatwość rozbudowy programowej i strukturalnej przedstawionego systemu symulacyjnego, jak również duża szybkość symulacji wynikająca z zastosowanej architektury systemu umożliwiają wykorzystanie systemu do celów:

- generacji programów skoordynowanej sygnalizacji świetlnej,
- realizacji łatwo dostępnego systemu wspomagania prac projektowych,
- weryfikacji organizacji ruchu na skrzyżowaniu oraz
- pracy symulatora w systemach czasu rzeczywistego dla oceny wyboru programów sterowania ruchem ulicznym.

LITERATURA

- [1] Praca zbiorowa: Pomiar i badania ruchu drogowego. Biblioteka Drogownictwa, WKŁ, Warszawa 1984.

- [2] Kaczmarek M.: Symulacja ruchu pojazdów w sieci ulicznej; Prace VII Krajowej Konferencji Automatyki. Rzeszów 1977.
- [3] Gault H.E., Taylor I.: Multiplan: a Computer Simulation Model of Multi-lane Signal-controlled Road during the Transition between Two Fixed Signal Plans; Transport Operation Reserarch Group, Research Report No. 41, University of Newcastle Upon Tyne, 1982.
- [4] Peterson J.: Petri Nets; Computer Surveys, Vol. 9, No 3, September 1977.
- [5] Praca zbiorowa: Symulacja ruchu potoku pojazdów. Biblioteka Drogownictwa. WKŁ, Warszawa 1980.
- [6] Kondratowicz L.: Modelowanie symulacyjne systemów. WNT, Warszawa 1978.
- [7] Praca zbiorowa: Modułowe systemy mikroprocesorowe. WNT, Warszawa 1984.
- [8] Nota reklamowa firmy ELWRO: ELWRO 800 - Systemy mikrokomputerowe serii ELWRO 800;
- [9] Praca zbiorowa: Badania operacyjne dla informatyków. WNT, Warszawa 1983.
- [10] Kompilator języka FORTRAN-CROOK dla minikomputera MERA-400 w systemie operacyjnym CROOK-4 - wersja 30/8, Poznań 1984.

Recenzent: Doc. dr hab. inż. Jan Piecha

СИМУЛЯЦИЯ УЛИЧНОГО ДВИЖЕНИЯ В МНОГОПРОЦЕССОРНОЙ СИСТЕМЕ
С ПРИМЕНЕНИЕМ МОДЕЛИ ОСНОВАННОЙ НА СЕТЯХ ПЕТРИ

Р е з ю м е

В статье представлено краткое описание сетей Петри и способ их использования для постройки симуляционной модели движения в сети улиц и перекрестков. Дана примерная модель движения транспорта на перекрестке для симуляции проводимой в многопроцессорной системе а также альтернативное решение выбранного элемента модели. Рассмотрена проблема разделений для отдельных подсистем в симуляционной системе с точки зрения эффективности работы всей системы.

Далее в статье показано использование сетей Петри анализа работы симуляционной системы, основанной на микрокомпьютерной многопроцессорной системе. Эта особенность позволила на учёт специфических черт компьютерного хардвера на этапе постройки модели а также на проведении выбора архитектуры системы для реализации поставленной перед ним задачи. С этой целью предложена постройка макромоделли т.е. сети Петри, описующей наиболее существенные свойства модели движения а также свойства многопроцессорной системы с предлагаемой архитектурой.

TRAFFIC SIMULATION IN MULTIPROCESSOR SYSTEM
USING A MODEL BASED ON PETRI NETS

S u m m a r y

In the paper we present a short description of Petri nets and a way of its use to build a simulation model of traffic in the street network with cross-roads. An exemplary model of a vehicle at a cross-road is presented suitable for multiprocessor system simulation. The alternative solution of a chosen element of the model is also presented. A problem of tasks distribution in the simulation system among different subsystems is discussed from the point of view of the efficiency of the whole system. The second part of the paper deals with the use of the Petri nets for the analyses of the simulation system based on the microcomputer multiprocessor system. It enables to take into account the peculiar characteristics of the used computer equipment in the stage of model design as well as the choice of the configuration of the system for the realization of the problem under consideration. Thus a macromodel has been proposed i.e. the Petri net which describes the most important features of the traffic model and the features of the multiprocessor system with proposed configuration.