XII Krajowa Konferencja
Teoria Obwodów i Układy Elektroniczne
Rzeszów - Myczkowce 17-20 X 1989

Referat F.03

Wojciech KALMAN
Bogdan WILAMOWSKI

Instytut Technologii Elektronicznej
Politechnika Gdańska

ELEKTRYCZNO-LOGICZNY SYMULATOR UKŁADÓW CYFROWYCH

Streszczenie: Zaprezentowano program symulatora układów cyfrowych opartego na algorytmie należącym do grupy algorytmó ładunkowych. Charakterystycznymi cechami programu są: możliwość analizy szerokiej klasy układów cyfrowych wykonanych różnymi technologiami oraz łatwość uwzględnienia wzajemnych połączeń między poszczególnymi funktorami.

ELECTRICAL-LOGICAL DIGITAL CIRCUITS SIMULATOR

Summary: Program for simulating asynchronic digital systems based on charge conservation algorithm is described. Many classes of digital circuits can be analyzed. Also effects caused by interconnection structure can be easy modelled.

1. WSTĘP

W ostatnich latach daje się zaobserwować wyraźny wzrost zainteresowania zastosowaniami techniki cyfrowej. Układy cyfrowe wkraczają do coraz szerszego użytku, z czego wynika konieczność szybkiego opracowania projektów i wdrożenia do produkcji wysoce niezawodnych układów zaspokajających zapotrzebowanie użytkowników. Wysoką niezawodność można zapewnić dzięki wykorzystaniu układów scalonych. Niezależnie od rodzaju projektowanego układu ("standard cells", "gate arrays" czy układy typu "full custom") jednym z najdroższych i najbardziej czasochłonnych etapów jest wytworzenie układu i wykonanie testów sprawdzających poprawność działania. W miarę wzrostu stopnia złożoności niezbędne staje się wykorzystanie narzędzi wspomagających proces projektowania. W celu eliminacji bardzo drogich i długotrwałych operacji wykonania i pomiarów struktur próbnych weryfikujących poprawność projektu i pozwalających na wykrycie i poprawienie błędów dąży się do opracowania narzędzi programowych służących powyższym zadaniom. Opracowany na Politechnice Gdańskiej program SUC-2 będący modyfikacją programów LOG4 [3] oraz SUC/SUGA daje wiele możliwości w tym zakresie. Oprócz sprawdzenia poprawności logicznej układu pozwala uwzględnić efekty pasożytnicze związane z pojemnościami obciążającymi funktory jak również łatwo analizować najgorszy przypadek (worst case). W poniższej pracy zawarto krótką charakterystykę algorytmu oraz opisano kolejne etapy działania programu SUC-2, który na komputerze klasy IBM XT/AT umożliwia analizę układów złożonych z ponad 5000 bramek. Ograniczenie to wynika ze specyfiki sposobu adresowania pamięci w systemach opartych na mikroprocesorach 8086/8088.

2. CHARAKTERYSTYKA ALGORYTMU

Programy analizy układów cyfrowych wykorzystują dwa podstawowe typy algorytmów. Używane są algorytmy elektryczne pozwalające na symulację układów opartą na matematycznym opisie modeli elementów układu. Są to programy zapewniające dużą dokładność wyników, ale ich zasadniczą wadą jest długi czas obliczeń i bardzo ograniczona wielkość analizowanych układów. Możliwa jest symulacja jedynie podstawowych komórek układu (max. kilkaset tranzystorów). Najczęściej spotykanym przedstawicielem tej grupy programów jest SPICE i jego modyfikacje. Typowymi dla układów cyfrowych są algorytmy analizy logicznej. Pozwalają one najczęściej jedynie na sprawdzenie poprawności realizacji funkcji logicznych układu. Wyniki uzyskane przy ich użyciu są najczęściej niewystarczające do oceny poprawności działania układu. Główną zaletą tej grupy programów jest ich duża szybkość i możliwość symulacji dużych układów. Często stosowane są algorytmy symulacji sterowanej zdarzeniami (tzw. event driven) operujące stałymi czasami opóźnień wnoszonych przez funktory. Podejście to wykorzystane między innymi w rozpowszechnionym programie OrCAD-VST, choć pozwala na wykrycie hazardów, nie jest w stanie uwzględnić efektów specyficznych dla układów scalonych uzależniających parametry dynamiczne od topografii.

Podstawowym założeniem podczas opracowywania programów LOG4, SUC/SUGA oraz SUC-2 było zapewnienie wiarygodności symulatora przy zachowaniu dużej szybkości działania. Program SUC-2 jest próbą osiągnięcia kompromisu między dokładnością uzyskiwanych wyników, szybkością symulacji oraz wielkością analizowanego układu. Oparty jest o algorytm należący do grupy algorytmów elektryczno-logicznych wykorzystujących zasadę zachowania ładunku. Model dynamiczny każdego funktora opisują następujące parametry:

- TRO - czas narastania napięcia wyjściowego nieobciążonej bramki
- TFO - czas opadania napięcia wyjściowego nieobciążonej bramki
- Co - pojemność wyjściowa bramki
- Ci - pojemność wejściowa bramki
- AL - amplituda logiczna
- IL - średni prąd wejściowy w czasie narastania impulsu wejściowego
- IH - średni prąd wejściowy podczas opadania impulsu wejściowego

Podczas symulacji funktor opisywany jest przez 4 efektywne parametry dynamiczne wyznaczone na podstawie złożonego modelu oraz efektów związanych z topografią układu:

- TR - efektywny czas narastania napięcia wyjściowego
- TF - efektywny czas opadania napięcia wyjściowego
- TDR - maksymalny czas opóźnienia przed narastaniem
- TDF - maksymalny czas opóźnienia przed opadaniem

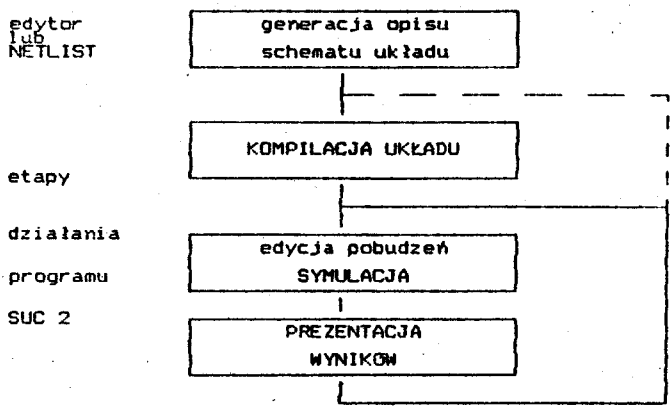
Na ich podstawie obliczane są maksymalny i minimalny ładunek zgromadzony w bramce oraz przyrosty ładunku wewnętrznego w pojedynczym kroku czasowym. Wartości parametrów TR, TF, TDR i TDF wyznacza się w oparciu o zależności:

$Ir = Co * AL / TRO$ - śr. prąd bramki w narastaniu
$If = Co * AL / TFO$ - śr. prąd bramki w opadaniu
$C = Co + \Sigma Ci + \Sigma Cz$ - pojemność węzła wyjściowego bramki
$TR = C * AL / (Ir + \Sigma IL)$ - efektywny czas narastania
$TF = C * AL / (If + \Sigma IH)$ - efektywny czas opadania

Algorytm opisany powyżej jest zmodyfikowaną wersją algorytmu przedstawionego w pracach [1][2]. Modyfikacje mają na celu przyspieszenie szybkości działania, zmniejszenie zajętości pamięci komputera podczas obliczeń oraz zapewnienie stabilności algorytmu przy symulacji z dużym krokiem czasowym.

3. OPIS DZIAŁANIA PROGRAMU

Wykorzystywanie programu SUC-2 może odbywać się według schematu przedstawionego poniżej. Można wyróżnić 3 zasadnicze etapy: kompilację układu, symulację oraz prezentację wyników. Program posiada dodatkowe możliwości takie jak zmiana kroku i czasu analizy, kompilacja i dekompilacja bibliotek zapewniające większą elastyczność systemu.



Rys 1. Schemat pracy z programem SUC-2

3.1. GENERACJA OPISU SIECI POŁĄCZEŃ

Program przystosowany jest do współpracy z popularnym edytorem schematów OrCAD - SDT. Dane wejściowe wprowadzane do programu SUC 2 powinny być zgodne z formatem INTERGRAPH. Opis sieci połączeń uzyskuje się przy pomocy programu NETLIST wchodzącego w skład pakietu OrCAD lub tworzy się przy pomocy dowolnego edytora tekstów przy zachowaniu zgodności z regułami formatu INTERGRAPH, który wybrano ze względu na zwartość i przejrzystość.

3.2. KOMPILACJA UKŁADU

Opis sieci połączeń jest wczytywany przez program SUC-2 i podlega dwuetapowej kompilacji. W pierwszym etapie analizowane są układy tworzące schemat oraz topografia sieci połączeń między nimi. W drugim etapie każdy układ jest rozpakowywany na tworzące go pojedyncze funktory. Ma tym etapie wykorzystywane są biblioteki modeli układów. Poprzednie wersje wykorzystywały prostą strukturę bibliotek, które zawierały opisy poszczególnych układów w formie "rozpakowanej" tzn. złożone z pojedynczych funktorów. Były to zbiory o dużej długości, co utrudniało ich użycie. Obecna wersja optymalizuje wykorzystanie bibliotek, które na czas dekompilacji ładowane są do pamięci. Dla minimalizacji ich rozmiarów oraz dla zwiększenia szybkości przetwarzania stworzono hierarchiczną strukturę zbiorów bibliotecznych. Przyjęto, że dane przechowywane są jako ciąg bajtów o następującym znaczeniu:

```
b1 - liczba wyprowadzeń układu
b2 - ilość funktorów tworzących układ
b3 - ilość makromodeli w układzie
     b4  - ilość końcówek 1 makromodelu
     b5  = 255 - identyfikator makromodelu   występuje
     b6  - długość nazwy                     b3
     b7 ÷ bn - kolejne znaki nazwy           razy
     bn+1 ÷ bm - numery kolejnych końcówek
bk     - ilość wejść pierwszego funktora
bk+1   - typ 1 funktora(1-NAND itd.)         występuje
bk+2   - numer modelu dynamicznego           b2
bk+3   - numer węzła wyjściowego             razy
bk+4 ÷ bp - numery kolejnych wejść
```

Po napotkaniu identyfikatora makromodelu w opisie układu następuje skok do miejsca w bibliotece, wskazanego przez zbiór dekodujący, odpowiadającego początkowi opisu tego makromodelu. Jeśli w opisie makromodelu pierwszego poziomu zostanie napotkany kolejny makromodel wykonywany jest kolejny skok na niższy poziom. Po rozpakowaniu makromodelu najniższego poziomu (możliwe jest zagnieżdżenie na max. 6 poziomach) następuje przeskok na poziom wyższy. Kolejne skoki powtarzają się do momentu całkowitego rozpakowania modelu układu. Następnie rozpakowywany jest kolejny układ wchodzący w skład schematu. Po zakończeniu rozpakowywania uzyskuje się strukturę złożoną z pojedynczych funktorów. Program pozwala na wykorzystanie do tworzenia modeli układów cyfrowych następujących funktorów: NAND, NOR, AND, OR, INV, XOR, NXOR, BUF.

W programie SUC-2 usunięto podstawową wadę uniemożliwiającą efektywną analizę układów cyfrowych CMOS. Poprzednie wersje nie miały zaimplementowanego modelu bramki transmisyjnej (TRAN), która jest jednym z podstawowych elementów układów cyfrowych wykonanych tą technologią, oraz bramek typu open collector (oznaczanych końcówką -O, np. OR-O).

Po rozpakowaniu całego układu cyfrowego i ewentualnym uwzględnieniu pojemności obciążających wyjścia bramek następuje obliczenie efektywnych parametrów każdego z funktorów. Zastosowanie symulacji układu na poziomie bramek, jak również obliczanie efektywnych parametrów dynamicznych każdego funktora pozwala zrealizować podstawowe założenie przyjęte we wstępnej fazie opracowywania programu, jakim było zapewnienie wiarygodności symulatora. Po wykonaniu obliczeń można przejść do kolejnego etapu pracy: definiowania pobudzeń i symulacji działania układu.

3.3. EDYCJA POBUDZEŃ I SYMULACJA

Parametry pobudzeń wprowadzane są przy użyciu zintegrowanego z programem modułu edycji ekranowej. Mogą one być zapisane do plików dyskowych, co ułatwia ich późniejsze wykorzystanie. Po zdefiniowaniu pobudzeń następuje zapis informacji o strukturze badanego układu do pliku informacyjnego i następuje symulacja. W każdym kroku czasowym analizowane są kolejne funktory. Na początku badane są bramki bezinercyjne (tzw. logiczne, oznaczane przedrostkiem L-, np. L-INV), w dalszej kolejności funktory inercyjne i bramki transmisyjne. Po zakończeniu analizy wszystkich funktorów rozpoczyna się symulacja w następnym kroku czasowym.

W każdym kroku czasowym zapisywane są w binarnym zbiorze dyskowym wyniki analizy. Symulacja złożonych układów prowadzi do wygenerowania dużej ilości danych wyjściowych, co narzuca większe wymagania na pamięć zewnętrzną. Aby umożliwić pracę z programem na podstawowej konfiguracji komputera typu XT (bez twardego dysku) przyjęto specyficzny format zbioru danych minimalizujący jego rozmiary, oraz zasadę zapisu tylko punktów rozpoczęcia i zakończenia procesu narastania napięcia w węzłach.

3.4. PREZENTACJA WYNIKÓW

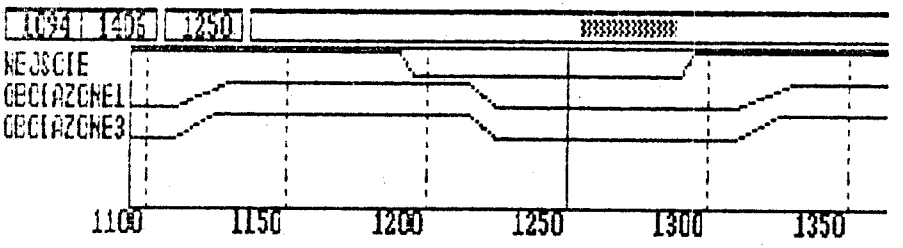
Dla prezentacji wyników analizy pracy układu opracowany został moduł edytora graficznego, który może być użyty zarówno jako opcja programu SUC-2 lub jako program samodzielny, gdyż komunikacja z programem symulacyjnym utrzymana jest na poziomie plików dyskowych. Moduł edycji graficznej pozwala na jednoczesne śledzenie do 20 przebiegów napięciowych w dowolnie wybranych węzłach układu. Program umożliwia łatwą zmianę skali na osi czasu, można również wybrać dowolny podprzedział czasu analizy. Precyzyjną obserwację zależności czasowych między przebiegami ułatwia ruchomy kursor, którego aktualna współrzędna jest stale wyświetlana na ekranie. W każdej chwili możliwe jest uzyskanie kopii wykresów z ekranu na drukarce. Wydruk zawiera dodatkowe informacje o układzie oraz, w celu usprawnienia pracy, o czasie wykonania symulacji.

3.5. POMOCNICZE OPCJE PROGRAMU

Program SUC-2 daje możliwość przetwarzania bibliotek zawierających opisy struktury modeli układów. Istnieje możliwość dekompilacji bibliotek binarnych (bezpośrednio wykorzystywanych) do postaci jawnej - tekstowej, w której można wprowadzać zmiany. Podczas kompilacji automatycznie tworzone są wszystkie niezbędne zbiory: z opisem struktury, dekodujący z adresami opisów poszczególnych struktur oraz zbiór modeli dynamicznych funktorów w danej bibliotece. Postać tekstowa jest bardzo uproszczona, zawiera nazwę układu, parametry modeli dynamicznych użytych bramek oraz opis sieci połączeń makromodeli i funktorów tworzących strukturę układu. Opcje powyższe w połączeniu z możliwością zmiany wykorzystywanej biblioteki pozwalają na analizę szerokiej klasy układów cyfrowych wykonanych różnymi technologiami.

4. PODSUMOWANIE

Poniższy rysunek przedstawia kopię fragmentu ekranu z postprocesora.



Rys 2. Kopia ekranu podczas pracy postprocesora

Poniższa tabela zawiera porównanie czasów analizy kilku prostych układów wykonanych przy pomocy programów SPICE i SUC-2.

układ	SPICE	SUC 2
oscylator 5inv	ok. 5h	6' 10"
oscylator 11inv	ok. 10h	10' 45"
przerzutnik D	ok. 22h	7' 50"
multiplexer 74157	—	15' 30"

Program SUC-2 jest nieco wolniejszy od poprzednich wersji, co wynika ze zwiększenia ilości poziomów logicznych (wprowadzono między innymi stan wysokiej impedancji). Wyniki uzyskane w programie SUC-2 pozostają w doskonałej zgodności z symulacją przy użyciu programu SPICE. Program pozwala wykryć przypadki występowania hazardów oraz wzbudzania się układów.

Bardzo wyraźny zysk czasu analizy oraz możliwość symulacji układów zawierających ponad 5000 bramek, przy zachowaniu dużej dokładności czyni z programu SUC-2 użyteczne narzędzie wspomagające proces projektowania układów cyfrowych. Jego ważną zaletą jest możliwość uwzględnienia efektów pasożytniczych związanych z topografią układu scalonego spowodowanych pojemnościami wnoszonymi przez metalizację.

Dalsze prace nad usprawnieniem symulatora koncentrują się wokół zagadnień związanych z przyspieszeniem procesu obliczeń w szczególności nad pomijaniem podczas analizy nieaktywnych bramek. Usprawnienia wymaga także kontakt z użykownikiem, prace mają na celu stworzenie programu w pełni przyjaznego dla użytkownika.

5. LITERATURA

[1] Wilamowski B.: Efektywny algorytm analizy stanów przejściowych w układach scalonych. VIII Krajowa KTOiUE, wrzesień 1985.

[2] Wilamowski B., Hamilton D., Staszak Z.: CHARCO - IC transient analysis program. IEEE International Circuit and System Conference, San Jose, May 1986.

[3] Cholewiński D., Wilamowski B.: Symulator asynchronicznych układów cyfrowych. XI Krajowa KTOiUE, październik 1988.