

Dariusz STACHAŃCZYK
POLITECHNIKA ŚLĄSKA

Projektowanie złożonych systemów elektronicznych w postaci asynchronicznej – analiza układów i narzędzi

Dr inż. Dariusz STACHAŃCZYK

Ukończył studia na Wydziale Automatyki, Elektroniki i Informatyki Politechniki Śląskiej, pracę doktorską obronił w 2005 r. Obecnie adiunkt w Instytucie Elektroniki Politechniki Śląskiej. Jego zainteresowania naukowe obejmują projektowanie komponentów wirtualnych, wykorzystanie wirtualnych komponentów do realizacji układów SoC oraz rozproszone środowiska projektowania.

e-mail: dariusz.stachanczyk@polsl.pl



Streszczenie

Pomimo złożoności problemu syntezy i ograniczonej dostępności narzędzi wspomagających proces projektowania potencjalne korzyści wynikające z zastosowania architektury asynchronicznej powodują, że problemem realizacji tego typu układów interesuje się coraz więcej ośrodków badawczych na całym świecie. W artykule przedstawiono wybrane zagadnienia związane z projektowaniem systemów asynchronicznych oraz ich najważniejsze zalety w stosunku do układów synchronicznych. Przedstawiona została również analiza wybranych asynchronicznych procesorów oraz przegląd dostępnych narzędzi wspomagających projektowanie układów asynchronicznych.

Słowa kluczowe: projektowanie układów asynchronicznych, projektowanie procesorów asynchronicznych, asynchroniczne komponenty wirtualne

Design of complex electronic systems in the asynchronous form – circuits and tools analyse

Abstract

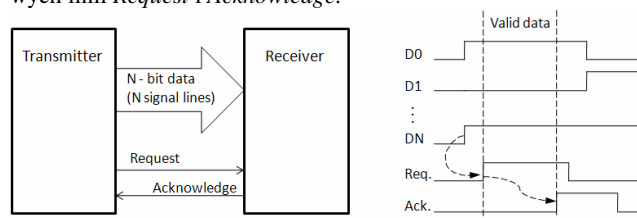
The substantial complexity of the synthesis problem and limited selection of tools supporting asynchronous circuit design caused that most of the designed nowadays electronic systems is implemented in a synchronous form. As distinct from synchronous circuits, that use clock signal to synchronise data flow, in a case of asynchronous circuits the implementation of an additional logic to control data flow is required. This causes that design of an asynchronous circuit is more problematic and the fabricated chip occupies more silicon area. On the other hand, the asynchronous implementation of the complex electronic system can bring several significant advantages over synchronous circuits. The most important asynchronous circuits features are low power consumption, less emission of electromagnetic noise and high operation speed. Therefore, despite an increased design complexity, more and more research and design groups are interested in asynchronous circuit design nowadays. The carried out research are focused mainly on the design of asynchronous processors, that could have a performance equal or better than synchronous equivalents and consume much less energy at the same time. The most important, existing implementations of the different asynchronous processors are discussed in this paper. Because the described processors were usually designed manually, mostly at transistor level thus its designs were very complex and time consuming. Since the choice of the efficient EDA tools supporting asynchronous circuit design is still very limited, some relevant examples of the existing tools are also presented within the paper.

Keywords: asynchronous circuit design, asynchronous processor design, asynchronous virtual components

1. Wstęp

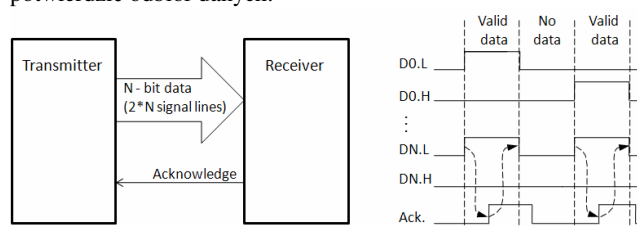
Założenie dyskretności czasu, którego odcinki wyznacza wspólny sygnał zegarowy doprowadzony do poszczególnych komponentów układu, istotnie upraszcza proces syntezy układów elektronicznych. Dlatego też zdecydowana większość realizowanych obecnie systemów elektronicznych projektowana jest w postaci układów synchronicznych. W odróżnieniu od układów

synchronicznych, w których przepływ danych pomiędzy komponentami synchronizowany jest sygnałem zegarowym, w przypadku układów asynchronicznych konieczna staje się implementacja dodatkowych mechanizmów sterujących przepływem danych pomiędzy poszczególnymi komponentami systemu. Zwykle wymiana danych pomiędzy komponentami systemu asynchronicznego realizowana jest za pomocą wybranego protokołu transmisji danych z potwierdzeniem (ang. *handshaking*). Do najczęściej wykorzystywanych rozwiązań należą protokoły *bundled-data* oraz *dual-rail*. W pierwszym przypadku (Rys. 1) każdy bit przesyłanej danej wymaga tylko jednej linii sygnałowej, ale do synchronizacji przepływu danych konieczne jest wprowadzenie dwóch dodatkowych linii *Request* i *Acknowledge*.



Rys. 1. Transmisja danych z potwierdzeniem - protokół *bundled-data*
Fig. 1. Handshake data transfer – *bundled-data* protocol

W przypadku protokołu *dual-rail* (Rys. 2) każdy bit danej reprezentowany jest za pomocą dwóch linii sygnałowych osobno dla wartości zero i jeden. Stan aktywności jednej z linii oznacza odpowiednio wartość zero lub jeden bitu danej, ale gdy obydwie linie są nieaktywne to dane są jeszcze niekompletne (nie ważne). Protokół ten wymagać może również zastosowania dodatkowego sygnału *Acknowledge*, za pomocą którego odbiornik mógłby potwierdzić odbiór danych.



Rys. 2. Transmisja danych z potwierdzeniem - protokół *dual-rail*
Fig. 2. Handshake data transfer – *dual-rail* protocol

Konieczność implementacji dodatkowych mechanizmów synchronizacji przepływu danych komplikuje proces syntezy układów asynchronicznych, a uzyskany w jej wyniku układ jest zwykle bardziej złożony niż jego synchroniczny odpowiednik. W efekcie do realizacji układu asynchronicznego niezbędna jest zwykle większa powierzchnia krzemu. Z drugiej strony jednak, realizacja systemu elektronicznego w postaci układu asynchronicznego niesie za sobą szereg potencjalnych korzyści, które byłyby trudne do osiągnięcia w przypadku systemu synchronicznego. Do najważniejszych zalet architektury asynchronicznej w porównaniu z układami synchronicznymi zaliczyć należy przede wszystkim [1][2]:

- brak problemów z dystrybucją globalnego sygnału zegarowego - W przypadku złożonych systemów elektronicznych realizowanych w postaci układów synchronicznych istotnym problemem jest takie rozprowadzenie sygnału zegarowego, aby zminimalizować różnice opóźnień sygnału docierającego do różnych fragmentów układu (skew), co mogłoby powodować nieprawidłowe działanie systemu. W odróżnieniu od układów synchronicznych opóźnienia sygnałów synchronizujących przepływ danych w układach asyn-

chronicznych nie mają tak krytycznego wpływu na działanie układu.

- *niski pobór energii* - Ze względu na brak globalnego sygnału zegarowego, w układach asynchronicznych nie trzeba stosować dodatkowych buforów i driverów linii zegarowych wykorzystywanych do eliminacji zjawiska skew, a energia nie jest marnowana na nieużyteczne zmiany stanu układu. Dodatkowo, zarówno cały układ jak również jego poszczególne komponenty składowe nie pobierają niepotrzebnie energii w przypadku kiedy nie pracują i nie przetwarzają danych.

- *niska emisja zakłóceń elektromagnetycznych* - W przypadku układów asynchronicznych wszelkie zmiany stanu układu wyzwalane są przez sygnał zegarowy, a tym samym zmiany stanu elementów układu następują jednocześnie i powtarzają się z częstotliwością równą częstotliwości sygnału zegarowego. W związku z tym, pobór energii występuje w wąskim paśmie wokół częstotliwości sygnału zegarowego i jej harmonicznych, co przekłada się na wysoki poziom zakłóceń elektromagnetycznych emitowanych z tymi częstotliwościami. Ponieważ zmiany stanu sygnałów synchronizujących przepływ danych w układach asynchronicznych występują w różnych, nieskorelowanych ze sobą odcinkach czasu widmo energii jest rozciągnięte w szerszym zakresie częstotliwości a maksymalna amplituda zakłóceń jest niższa niż w przypadku układów synchronicznych.

- *wysokie prędkości pracy* - Pomimo faktu, że dla różnych danych wejściowych czas ich przetwarzania przez poszczególne elementy układu może być różny, maksymalna częstotliwość sygnału zegarowego układu synchronicznego musi być tak dobrana aby zapewnić prawidłową pracę w najgorszym przypadku, czyli w sytuacji najdłuższego czasu przetwarzania danych. Prędkość pracy układu asynchronicznego może zmieniać się natomiast dynamicznie w zależności od danych wejściowych, dlatego też ogólna prędkość pracy układu wynika z średniej prędkości pracy poszczególnych komponentów układu.

- *większe możliwości modularyzacji układu* - Brak problemów związanych z koniecznością synchronizacji sygnałów zegarowych poszczególnych komponentów składowych systemu, integracja komponentów asynchronicznych jest prostsza niż w przypadku układów synchronicznych i ogranicza się przede wszystkim do dopasowania interfejsu transmisji danych z potwierdzeniem i lokalnych zależności czasowych pomiędzy komponentami.

- *większe możliwości modularyzacji układu* - Ponieważ prędkość pracy układów VLSI jest zależna od warunków w jakich one pracują, np. temperatury czy napięcia zasilania układu, poprawna praca układu synchronicznego przy zadanej częstotliwości zegara taktującego zagwarantowana jest zwykle tylko dla określonego zakresu wartości wybranych parametrów. Architektura układów asynchronicznych zapewnia natomiast dużo większą odporność na zmianę warunków pracy czy parametrów procesu technologicznego i umożliwia adaptację do innych warunków poprzez zmniejszenie lub zwiększenie prędkości pracy układu wtedy, kiedy jest to konieczne.

2. Asynchroniczne procesory - przykłady

Pomimo istniejących ograniczeń i trudności w realizacji złożonych systemów asynchronicznych, możliwość uzyskania wymierzonych korzyści wynikających z wykorzystania architektury asynchronicznej powoduje, że problematyką projektowania tych układów interesuje się obecnie coraz więcej ośrodków badawczych. Prowadzone badania skoncentrowane są przede wszystkim na realizacji rdzeni mikroprocesorów, które dzięki asynchronicznej architekturze przy znacznie niższym poborze energii osiągałyby wydajność równą lub nawet większą od ich synchronicznych odpowiedników. Jednym z pierwszych opisywanych w literaturze mikroprocesorów o architekturze asynchronicznej jest zrealizowany w 1988 roku *Caltech Asynchronous Microprocessor* (CAM) [3]. Według informacji opublikowanych przez jego projektantów, ten szesnasto-bitowy procesor o architekturze RISC wykonany w technologii 1.6µm osiągał wydajność od 5 MIPS przy zasilaniu

napięciem 2V i poborze prądu 5,2mA do 26 MIPS przy zasilaniu napięciem 10V i poborze prądu 105mA. W wyniku dalszych prac poświęconych realizacji złożonych układów asynchronicznych w latach 1995-1998 w laboratorium w Caltech opracowany został kolejny projekt - procesor MiniMIPS będący asynchroniczną wersją procesora MIPS R3000 [4]. Zrealizowany system składał się z 32-dwu bitowej jednostki centralnej RISC oraz układu zarządzania pamięcią. Cały system poddany został odpowiedniej dekompozycji na bloki, a następnie elementarne komórki funkcjonalne, których architektura była projektowana ręcznie na poziomie tranzystorów. Według przeprowadzonych symulacji zaprojektowany procesor mógł potencjalnie osiągnąć wydajność 280 MIPS przy zasilaniu 3,3V. Pomimo błędów w projekcie i ograniczeń procesu technologicznego, zrealizowane w technologii CMOS 0.6µm pierwsze prototypy procesora, osiągały zadawalającą wydajność około 180 MIPS przy zasilaniu 3,3V pobierając jednocześnie 4W, potwierdzając tym samym zalety wynikające z wykorzystania architektury asynchronicznej. Obiecujące wyniki uzyskane w przypadku procesorów CAM i MiniMIPS, szczególnie niski poziom pobieranej energii, spowodowały, że kolejnym celem grupy badawczej z Caltech stało się opracowanie energooszczędnej wersji mikroprocesora 8051 [5]. Bazując na wynikach przeprowadzonych symulacji, projektanci szacują, że układ wykonany w technologii CMOS 0.5µm powinien przewyższać pod względem osiąganych parametrów inne, dostępne na rynku implementacje procesora 8051.

Badania związane z realizacją złożonych układów asynchronicznych prowadzone są również przez grupę badawczą APT (Advanced Processor Technologies) na uniwersytecie w Manchester [6]. W wyniku prowadzonych prac powstała seria procesorów AMULET będących asynchronicznymi odpowiednikami 32-dwu bitowych procesorów ARM. Podstawowym celem projektów w wyniku których powstały pierwsze wersje procesorów AMULET (AMULET1 i AMULET2e) była demonstracja możliwości praktycznej realizacji złożonych systemów elektronicznych w postaci układów asynchronicznych. W przypadku trzeciej wersji procesora AMULET (AMULET3) za główny cel obrano sobie realizację asynchronicznego procesora zgodnego z architekturą ARM9, który mógłby konkurować pod względem wydajności i energooszczędności z najnowszym, dostępnym w tym czasie, synchronicznym procesorem ARM9TDMI. W 2000 roku wyprodukowany został prototyp asynchronicznego układu SoC, w którym w roli jednostki centralnej użyto procesor AMULET3. Wykonany w technologii CMOS 0.35µm procesor osiągał wydajność 100 MIPS, a zużycie energii wyrażone stosunkiem wydajności do pobieranej mocy wynosiło 620 MIPS/W [7].

Kolejnym przykładem udanej implementacji procesora o architekturze asynchronicznej jest opracowany na uniwersytecie w Tokio procesor TITAC-2 [8]. Wyprodukowany w 1997 roku w technologii CMOS 0.5µm TITAC-2 zaprojektowany został w oparciu o architekturę 32-dwu bitowego procesora MIPS R2000. Przeprowadzone testy prototypu potwierdziły zdolność układów asynchronicznych w zmiennych warunkach otoczenia - TITAC-2 pracował prawidłowo w szerokim zakresie napięć zasilania (1,5÷6V) i temperatur (196÷100°C) oraz osiągał wydajność 54 VAX MIPS przy zasilaniu 3,3V i poborze energii wynoszącym 2W.

O ile przedstawione powyżej przykłady potwierdzają, że możliwa jest skuteczna implementacja asynchronicznych wersji mikroprocesorów, o tyle złożoność i czasochłonność, w większości przypadków realizowanego manualnie na poziomie tranzystorów projektu może istotnie ograniczyć zainteresowanie projektantów tego typu układami. Dlatego też, opracowanie syntezowalnego, asynchronicznego modelu procesora, który mógłby zostać udostępniony szerokiemu gronu projektantów w formie darmowego wirtualnego komponentu stało się celem projektu ASPIDA [9]. W efekcie zaprojektowany został w pełni syntezowalny, opisany w języku Verilog, model asynchronicznej wersji procesora DLX, który może być wykorzystany w zarówno w projektach implementowanych w układach FPGA jak i ASIC.

Znaczące korzyści wynikające z wykorzystania architektury asynchronicznej spowodowały, że poza projektami powstającymi w ramach prowadzonych prac naukowych realizowane są w chwili obecnej również rozwiązania komercyjne. Obecnie na rynku europejskim są dwie liczące się firmy - ARM oraz Handshake Solutions – posiadające w swojej ofercie rdzenie asynchronicznych mikroprocesorów. Handshake Solutions [10] oferuje w postaci wirtualnych komponentów asynchroniczne rdzenie mikrokontrolerów kompatybilnych z układem INTEL 80C51. Zaprojektowane z wykorzystaniem opracowanych przez Handshake Solutions narzędzi do syntezy układów asynchronicznych mikrokontrolery HT80C51 i HT80C51MX, wykonane w technologii CMOS 0.14μm osiągają odpowiednio wydajność 12 i 16 MIPS przy zasilaniu 1,8V. Jednocześnie komponenty te charakteryzują się bardzo małym poborem energii i niską emisją zakłóceń elektromagnetycznych. Narzędzia i doświadczenia projektowe Handshake Solutions wykorzystane zostały przez firmę ARM do realizacji pierwszego asynchronicznego procesora oferowanego przez ARM – układu ARM996HSTM [11]. Ze względu na bardzo mały pobór energii oraz niską emisję zakłóceń elektromagnetycznych procesor ARM996HSTM przeznaczony jest przede wszystkim do zastosowań w przemyśle motoryzacyjnym, urządzeniach medycznych czy wbudowanych systemach sterowania.

3. Narzędzia projektowe

Jednym z najważniejszych czynników wpływających na niską popularność układów asynchronicznych jest ograniczona dostępność wydajnych narzędzi projektowych. Przedstawione w poprzednim rozdziale układy projektowane były w większości przypadków ręcznie, zwykle na poziomie pojedynczych tranzystorów tak, aby uzyskać niski pobór mocy lub dużą wydajność układu. Wykonanie w ten sposób prawidłowo działającego układu wymaga dużego doświadczenia projektantów i znacznych nakładów czasu. Dodatkowo, możliwość wykorzystania w ten sposób zrealizowanych układów w innych projektach jest bardzo ograniczona. Powyższe ograniczenia mogłyby zostać wyeliminowane poprzez zastosowanie wydajnego języka opisu sprzętu i narzędzi syntezy pozwalających na implementację układów asynchronicznych.

W wyniku badań prowadzonych od 1986 roku przez Philips Research Labs w Eindhoven opracowany został język Tangram mający usprawnić proces projektowania układów asynchronicznych o niskim poborze mocy [12]. W tym celu, poza językiem opisu sprzętu opracowany został również kompilator umożliwiający syntezę opisanych w języku Tangram układów oraz zestaw dodatkowych narzędzi pozwalających na ocenę wydajności i testowanie realizowanych systemów a synchronicznych. Używany i rozwijany przez Philips Semiconductors system Tangram stał się bazą do realizacji środowiska projektowania systemów asynchronicznych TiDETM (Timeless Design Environment) opracowanego przez firmę Handshake Solutions. Oferowane obecnie komercyjnie narzędzia Handshake Solutions wykorzystane i zweryfikowane zostały w trakcie realizacji przedstawionych w poprzednim rozdziale wirtualnych komponentów procesorów 8051 i ARM.

Alternatywą dla Tangram jest opracowane na uniwersytecie w Manchester środowisko Balsa [2][13]. Analogicznie jak w przypadku systemu środowiska Tangram, język opisu Balsa oraz dołączone narzędzia do syntezy pozwalają na kompilację funkcjonalnego opisu układu asynchronicznego do postaci komponentów składowych komunikujących się za pomocą wybranego protokołu transmisji danych z potwierdzeniem (*handshaking*). Projektanci grupy APT (Advanced Processor Technologies) na uniwersytecie w Manchester wykorzystali środowisko z Balsa w projekcie syntezywalnego modelu kolejnej wersji procesora AMULET [14]. Wykorzystanie narzędzi do syntezy nie pozwoliło co prawda uzyskać tak dobrych parametrów układu jak to miało miejsce w przypadku poprzednich, ręcznie projektowanych wersji tego procesora, ale istotnie przyspieszyło proces implementacji systemu.

4. Podsumowanie

Przedstawione w rozdziale pierwszym potencjalne korzyści, które niesie za sobą realizacja systemu elektronicznego w wersji asynchronicznej mogą w zależności od realizowanej aplikacji mieć mniejsze lub większe znaczenie. Konieczność rozbudowy układu o dodatkowe elementy odpowiedzialne za kontrolę przepływu danych może przełożyć się na takie zwiększenie powierzchni wykorzystanego krzemu czy pobieranej mocy, że wykorzystanie architektury asynchronicznej może okazać się mało opłacalne. Nie mniej jednak, analiza wyników uzyskiwanych przez projektantów asynchronicznych wersji procesorów przedstawionych w rozdziale drugim wskazuje, że możliwe jest uzyskanie wymiernych korzyści, szczególnie w zakresie pobieranej przez układ energii. Niestety, istotne utrudnienie stanowi wciąż brak odpowiednich narzędzi EDA, które wspierałyby proces projektowania i testowania układów asynchronicznych, a tym samym pozwalały na utrzymanie na akceptowalnym poziomie kosztów i czasu realizacji tego typu układów.

Chociaż problemem projektowania i implementacji układów asynchronicznych interesuje się od kilku lat szereg ośrodków akademickich jak i przemysłowych na całym świecie, niestety w chwili obecnej poza narzędziami wymienionymi w rozdziale trzecim nie udało zidentyfikować się innych istotnych rozwiązań wspomagających proces projektowania tego typu układów. O ile, rozwiązania te opracowane i weryfikowane były przede wszystkim pod kątem implementacji asynchronicznych układów ASIC, o tyle brakuje informacji o ich potencjalnym wykorzystaniu do implementacji układów asynchronicznych w strukturach programowalnych CPLD i FPGA. Dlatego też, w ramach dalszych prac badawczych autora, prowadzone będą doświadczenia mające na celu weryfikację potencjalnych zalet układów asynchronicznych w przypadku ich implementacji w strukturach programowalnych.

5. Literatura

- [1] Chris J. Myers: Asynchronous Circuit Design. John Wiley & Sons Inc., 2001.
- [2] Jens Sparsø, Steve Furber: Principles of Asynchronous Circuit Design - A Systems Perspective. Kluwer Academic Publishers, 2001.
- [3] Alain J. Martin: The Design of an Asynchronous Microprocessor. Technical Report. California Institute of Technology.
- [4] Alain J. Martin, et al.: The Design of an Asynchronous MIPS R3000 Microprocessor. Proceedings of the 17th Conference on Advanced Research in VLSI, IEEE Computer Society Press, 1997.
- [5] Alain J. Martin, et al.: Lutonium: A Sub-Nanojoule Asynchronous 8051 Microcontroller. Proc. of the 9th IEEE International Symposium on Asynchronous Systems & Circuits (ASYNC), 2003.
- [6] The Advanced Processors Technologies Group, University of Manchester, <http://intranet.cs.man.ac.uk/apt/>.
- [7] J.D. Garside, et al.: AMULET3i - an Asynchronous System-on-Chip. Proceedings of the 6th International Symposium on Advanced Research in Asynchronous Circuits and Systems, 2000.
- [8] Akihiro Takamura, Masashi Kuwako, Masashi Imai, et al.: "TITAC-2: A 32-bit Asynchronous Microprocessor based on Scalable-Delay-Insensitive Model" Proceedings of ICCD'97, 1997.
- [9] ASPIDA Project, IST-2002-37796, <http://www.ics.forth.gr/carv/aspida/>.
- [10] Handshake Solutions, <http://www.handshakesolutions.com/>.
- [11] ARM, ARM996HS processor, <http://www.arm.com>.
- [12] J. Kessels, A. Peeters: The Tangram framework: asynchronous circuits for low power. Proceedings of the ASP-DAC 2001, Design Automation Conference, 2001.
- [13] The Balsa Asynchronous Synthesis System, <http://intranet.cs.man.ac.uk/apt/projects/tools/balsa/>.
- [14] L.A. Plana, P.A. Riocreux, W.J. Bainbridge, et al.: SPA - A Synthesizable Amulet Core for Smartcard Applications. Proceedings of the 8th International Symposium on Asynchronous Circuits and Systems, 2002.