

**ИЗПОЛЗВАНЕ НА ТЕХНИКИ ОТ НИСКО НИВО ЗА ПРОЕКТИРАНЕ НА
ВИСОКО НИВО****HIGH-LEVEL DESIGN METHODOLOGY USING LOW-LEVEL TECHNIQUES****Петър Минев***Технически университет – Габрово***Валентина Куценска***Технически университет - Габрово***Abstract**

While it is considered that the transition from RTL to the next level of ESL design will radically change the design of electronic devices and activities of most companies working in this area, it is still difficult and achievable languages and tools designed for this purpose remain without commercial success. Even now widely used methodologies based on design language, concentrated mainly on RTL level have some drawbacks, such as that typically use hardware description language does not correspond to the way designers think, along with hardware description language is necessary learning disciplines and techniques of software programming. In addition designers should know ways in which some constructions and operators affect the synthesis tools. Modern synthesis tools are relatively powerful and achieve good results, but they do not give the necessary degree of control over the project - code written in two ways that simulate uniformly on register-transfer level can be synthesized by radically different schemes at the gate-level. Therefore techniques from lower levels of abstraction, as schematic design, flow charts and state diagrams are still necessary to describe and implementation of the projects.

This article describes different techniques for designing low and a higher level of abstraction and the need to combine them in the design of digital circuits and devices.

Keywords: ключова дума на английски език; ключова дума на английски език.

ВЪВЕДЕНИЕ

През деветдесетте години на миналия век системното или поведенческо ниво на проектиране придобива важно значение, тъй като интегралните схеми се превръщат в системи сами по себе си, т. нар. вградени системи или SoC (System on Chip), при които в една интегрална схема се вграждат микропроцесор, памет, входно-изходни устройства и специализирани хардуерни модули. Всички тези устройства е необходимо да се моделират, за да се прецени общата системна производителност, а инструментите за системно проектиране позволяват да се изследват различни архитектурни варианти, за да се определи дали дадена функция да се изпълни от микропроцесор или не, т.е. дали да се направи софтуерно или хардуерно. За тази цел се използват възможностите, които дават VHDL и Verilog за моделиране на поведение, но също така в употреба навлизат и чисто софтуерни езици, като C и C++.

Досега все още няма отворен стандартен език или методика за системен дизайн. Опити в това отношение са SystemC и Superlog, но без комерсиален успех [1][2].

От друга страна използваните в момента методологии базирани върху HDL езиците за проектиране са концентрирани основно върху RTL нивото. Те имат следните недостатъци: използването на езици за хардуерно описание не отговаря на начина по който мислят проектантите; наред с езика за хардуерно описание е необходимо изучаване и на дисциплини и техники от софтуерното програмиране; проектанта трябва да познава начините по-които някой конструкции и оператори в езика влияят на инструментите за синтез. Съвременните инструменти за синтез са сравнително мощни и постигат добри резултати, те обаче не дават необходимата степен на контрол върху проекта – код написан по два начина, които се симулират по еднакъв начин на RTL ниво, могат да се синтезират до коренно различни схеми на гейтово ниво.

За избягване на изброените недостатъци са необходими допълнителни техники и методи за описание и реализация на проектите.

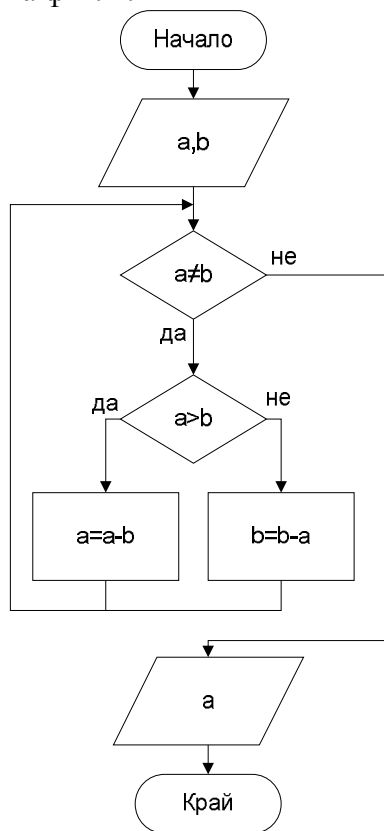
В настоящия доклад се използва RT-методологията за да се направи хардуерна имплементация на алгоритъм за намиране на

най-голям общ делител [3], като тя е съчетана със следните допълнителни методики: йерархично проектиране на база параметризирани модули; проектиране на база алгоритмични блок-схеми и диаграми на състоянията; схемотехничен дизайн на определени части от проекта.

При реализацията на проекта е използвана автоматизирана среда за разработка Xilinx ISE WebPack 7.1i и развойна платка Xilinx CoolRunner-II Starter Kit.

ИЗЛОЖЕНИЕ

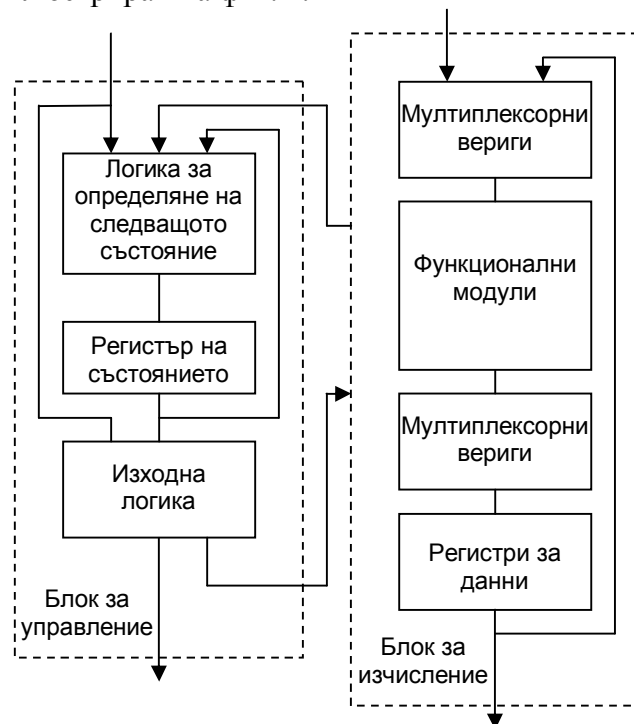
Блок-схемата на използвания алгоритъм за намиране на най-голям общ делител е показана на фиг. 1.



Фиг. 1. Блок-схема на алгоритма за намиране на най-голям общ делител

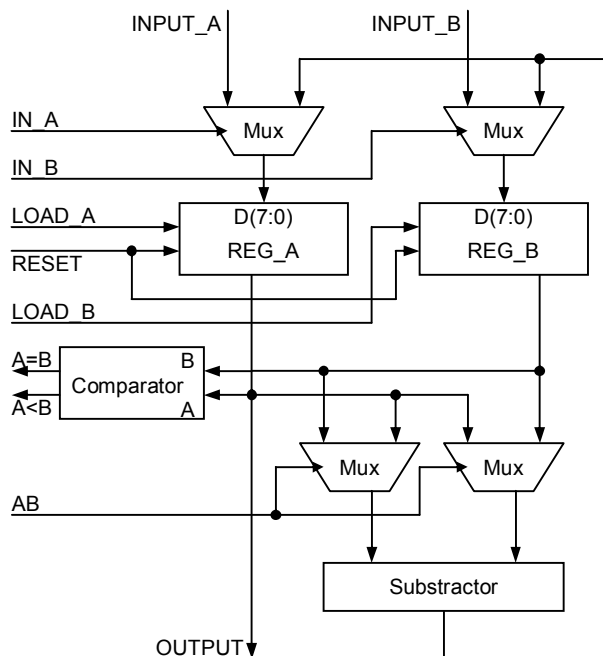
Методиката за хардуерна реализация на алгоритми на RTL ниво, наречена за краткост RT-методология [4][5], изисква разделяне на схемата на блок за управление (control unit) и блок за изчисление (datapath unit). Блока за управление обикновено се реализира като краен автомат на Мили, а в блока за изчисление участват схеми за аритметични операции, регистри за съхранение на временните стойности на променливите и мултиплексори за пренасочване на входовете и изходите на регистрите и схемите за

аритметични операции. Този принцип е илюстриран на фиг. 2.



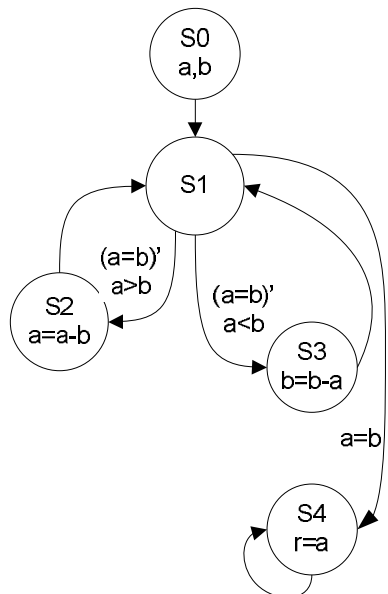
Фиг. 2. Блокова схема, представяща принципа на RT-методологията

На фиг. 3 е показана структурната схема на блока за изчисление, решаващ проблема за намиране на най-голям общ делител. В лявата част са изведени сигналите за връзка с блока за управление.



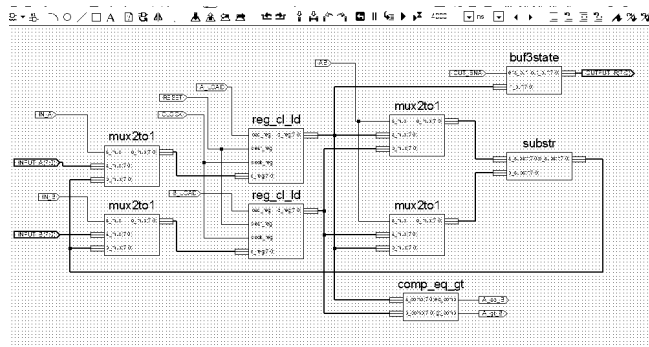
Фиг. 3. Структурна схема на блока за изчисление, намиращ най-голям общ делител

На фиг. 4 е дадена диаграмата на състоянията за крайния автомат, реализиращ блока за управление.

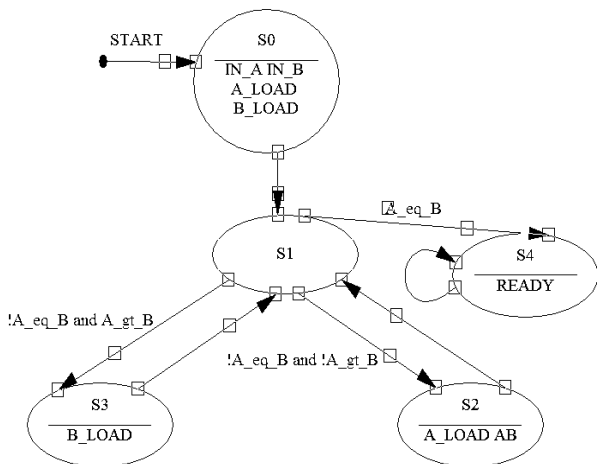


Фиг. 4. Диаграма на състоянията на крайния автомат, реализиращ блока за управление

За реализацията на блока за изчисление са използвани предварително създадени от авторите параметризирани модули [6], които се свързват в структурна схема, както е показано на фиг. 5. На база създадената схема се генерира структурен VHDL модел, който в последствие се използва при оформянето на проекта на най-високото йерархично ниво.

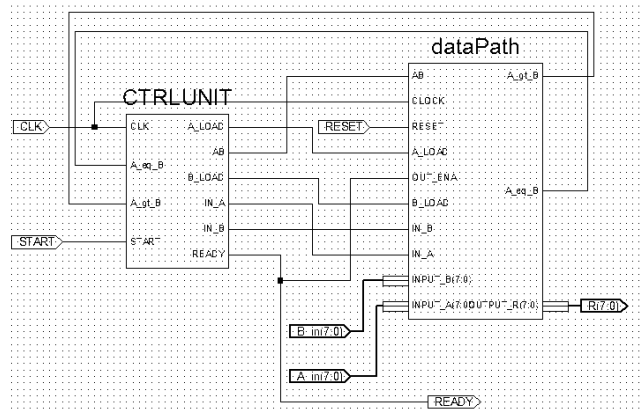


Фиг. 5. Реализацията на структурната схема на блока за изчисления



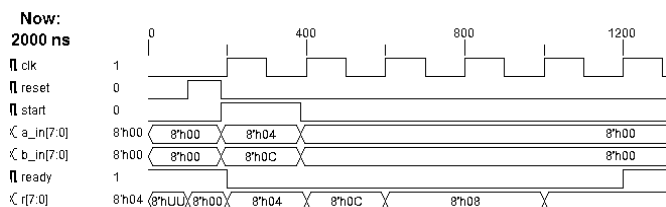
Фиг. 6. Диаграма на състоянията въведена в StateCAD

Крайният автомат на Мили също е изготвен като VHDL модул, като за целта е използван инструментът за автоматизирано проектиране на крайни автомати StateCAD [7]. Този инструмент приема като вход графично зададената диаграма на състоянията на автомата (фиг. 6) и генерира поведенчески VHDL модел, който участва в най-горното йерархично ниво на проекта.



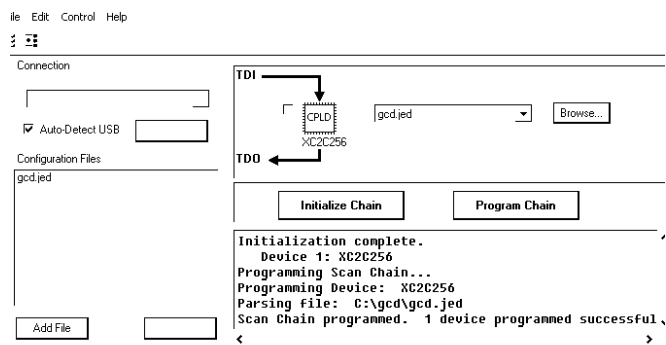
Фиг. 7. Най-горното йерархично ниво на проекта

На фиг. 7 е показано най-горното йерархично ниво, свързващо блока за управление и блока за изчисления. От това ниво се генерира структурен VHDL модел, който се синтезира до гейтово ниво и получената схема се имплементира в избраният програмируем прибор.



Фиг. 8. Резултати от симулацията на проекта

Резултатите от симулацията на проекта са показани на фиг. 8, а резултатът от конфигурирането на програмируемата логическа схема CPLD CoolRunner-II XC2C256 е показан на фиг. 9.



Фиг. 9. Успешното конфигуриране на проекта в избраната програмируема логическа схема

ЗАКЛЮЧЕНИЕ

В този доклад е описана една методика за проектиране на цифров хардуер, базирана на RTL-методологията, която включва и допълнителни техники и методи, използвани в съвременните среди за автоматизирано проектиране.

Методиката съчетава в себе си както подхода „отгоре-надолу”, разделяйки проекта на по-малки блокове, така и подхода „отдолу-нагоре”, използвайки предварително проектирани и тествани параметризирани модули за реализация на някои от блоковете.

При използване на описаната методика отпада необходимостта от детайлно познаване на конкретен език за проектиране, тъй като при описание и изпълнение на проектите се използват традиционни инструменти, като схемни редактори, графични редактори на диаграми на състоянията и библиотеки от готови модули, които лесно могат да се променят според нуждите на конкретния проект.

Изследването на генерирания от тези графични инструменти HDL код, може да послужи и за целите на обучението по даден език за описание на хардуер, като VHDL или Verilog.

Недостатъци на предложената методика в сравнение с традиционното HDL проектиране на RTL ниво са: липса на гъвкавост при формирането на HDL кода, поради използване на генерирани или готови модули; усложнени структурни и поведенчески модели, генерирани от инструментите за автоматизирано проектиране; слаба оптимизация и използване на повече ресурси от програмируемата логическа схема.

За избягване на изброените недостатъци е необходимо разработване на автоматизирана система, която комбинира техниките и методите описани по-горе и оптимизира генерираният HDL код на RTL ниво, така че да бъде ефективно синтезиран до ниво логически елементи.

Предложеният подход остава обект на бъдещи изследвания и разработки от страна на авторите.

ЛИТЕРАТУРА

- [1] Sweeney Charles, Hardware Design Methodologies. Celoxica Limited, 2002
- [2] Browy C., Gullikson G., Indovina M., *A Top-Down Approach to IC Design*. Draft v1.1, 1997

- [3] Hwang Enoch, Digital Logic and Microprocessor Design with VHDL. Thomson, 2006.
- [4] Chu Pong, RTL Hardware Design Using VHDL. A John Wiley & Sons, Inc., 2006.
- [5] Chu Pong, FPGA Prototyping by VHDL Examples. A John Wiley & Sons, Inc., 2008.
- [6] Минев П., В. Куценска, Библиотека от VHDL модели на елементи от компютърните системи, Сборник доклади от Международна научна конференция “УНИТЕХ’06”, Габрово, 24 – 25 Ноември, 2006, том I, стр. 330–335.
- [7] Симеонов И., Структурно моделиране и изследване на автомат на Мили в среда на WebPack, Сборник доклади от Международна научна конференция “УНИТЕХ’06”, Габрово, 24 – 25 Ноември, 2006, том I, стр. 312–316.