МЕТОДИ ЗА ДИЗАЙН ПРИ ПРОЕКТИРАНЕТО НА ЦИФРОВИ ИНТЕГРАЛНИ СХЕМИ

Благомир Росенов Дончев

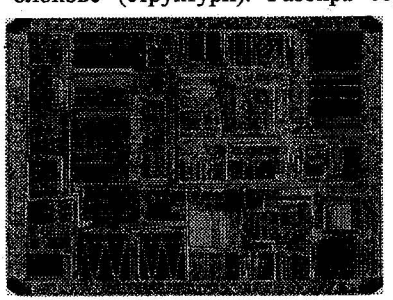
Технически Университет – София, ECAD лаборатория

donchev@ecad.vmei.acad.bg

Donchev B.R.,Methods for design of digital integrated circuits. *There are two major ways to approach any situation: "Top-Down" and "bottom-up". An understanding of how these two tactics complement each other can help reveal when each is most appropriate. In this paper are presented general aspects of these methods and compare their advantage and disadvantage. A "Top-Down" style moves from general to specific. It begins with the big picture, overarching principles, wide-angle views, and then zooms in to reveal increasing levels of fine detail. A "Bottom-Up" style, on the other hand, goes from atomic scales toward the macroscopic world. It begins with fundamentals, first principles, infinitesimal slivers, and then puts those building blocks together to create the whole. To improve quality of design, "Top-Down" methods are combining with hardware description languages (HDL). They are a software programming language used to model the intended operation of a piece of hardware. There are two aspects to the description of hardware that an HDL facilitates: true abstract behavior modeling and hardware structure modeling.*

1. Въведение

С увеличаване на сложността в съвременните дизайни и намаляване на времето им за разработка, проектантите все по трудно успяват да завършват успешно своята разработка без използването на подхода "Top-Down". Той се отличава с по-голяма простота и възможност за бързо изграждане на дизайна, използвайки предварително разработени блокове (структури). Разбира се, подхода "Top-Down" изисква разработката и следването на определени методи за проверка, оптимизация и методика на проектиране при трансформирането на дизайна от абстрактна блокова диаграма към детайлна реализация на ниво транзистори.

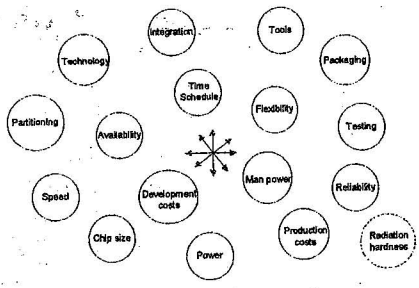


Фиг.1 Изглед на съвременен ИС дизайн

Днес аналогово-цифровите дизайни реализират сложни алгоритми, които изискват от дизайнерите да описват множество обработки с помощта на хиляди цикли. Пример за това са PLL схеми, сигма-делта преобразуватели, управление на PRML канали и CDMA приемо-предаватели.

При аналоговите дизайни, основна причина за лошата продуктивност е наличието на увеличаваща се сложност, комбинирана с продължително предпочитане на подхода за проектиране "bottom-up" (на транзисторно ниво). Тестването закъснява в етапа на проектирането, което води често до грешки и допълнителни усложнения. Съществува голяма разлика в продуктивността между дизайнерите, които са направили прехода към ефективният подход на проектиране "Top-Down" и използват езици за поведенческо описание от високо ниво (MS-HDL) и тези които все още практикуват "bottom up" подхода и средата Spice, за проверка на своята разработка.

2. Подход за проектиране "Bottom-Up"



Фиг.2 Условия за разработка

"Bottom-Up" дизайна започва с проектирането на отделни блокове които след това се комбинират в една обща работеща система. Проектирането на отделните блокове започва с уточняване на спецификацията и завършва с реализацията на транзисторно ниво, като характерното за този подход е че всеки блок се проверява като самостоятелен възел в зависимост от спецификацията, но не и като съставна част от една система. Когато етапа на проверка приключи, блоковете се комбинират след което се проверява работата на системата като цяло. От тази гледна точка, системата е представена на транзисторно ниво.

Докато "Bottom-Up" подхода продължава да бъде ефективен за малки дизайни, при големите дизайни възникват няколко важни проблема свързани с подхода:

- Симулацията на системата като цяло, отнема много време. Проверката е усложнена, дори по някога и невъзможна. Времето за проверка е необходимо да бъде намалено от гледна точка на времето за цялостното проектиране, като непълни проверки носят риска от допускане на грешки, които да доведат до забавяне на крайния дизайн.
- За сложни дизайни, постигането на добри характеристики, цена и функционалност са типични за архитектурното ниво. Чрез "Bottom-Up" подхода, изследването за това кой тип архитектура удовлетворява по-горе посочените цели, често липсва.
- Всички грешки или проблеми намерени когато системата вече е асемблирана са скъпи за отстраняване, тъй като отделните блокове са на транзисторно ниво.
- Няколко важни и скъпи стъпки в "Bottom-Up" подхода на проектиране, трябва да бъдат изпълнявани последователно, което увеличава времето,

необходимо за завършване на дизайна. Например проверката на системно ниво и разработката на тестове.

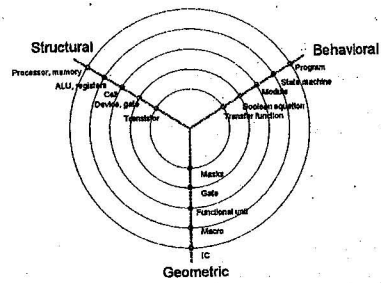
За да отговорят на това предизвикателство, много от дизайнерите се обръщат или вече използват "Top-Down" методологията на проектиране.

3. Подход за проектиране "Top-Down"

В основата на "Top-Down" подхода, архитектурата на чипа се дефинира като блок-диаграма, като симулациите и оптимизациите се извършват чрез използването MS-HDL симулатор или системен симулатор.

От симулациите на високо ниво се извличат изискванията за отделните съставни схемни блокове. След това се разработват отделните схемни реализации, съобразно вече установената спецификация. Накрая разработвания чип се изпитва като една цялостна система и резултатите се съпоставят с първоначалното задание.

Добре разработения "Top-Down" дизайн методично преминава от архитектурен дизайн към дизайн на транзисторно ниво. Всяко йерархично ниво е напълно завършено преди да се премине към по-горното. Това разделяне на дизайна на по-малки, добре дефинирани блокове, позволява на дизайнерите да работят заедно по-ефективно. Това от своя страна води до намаляване на цялостното време необходимо за разработка на дизайна. Намалява се броя на пропуските допускани по време на проектирането и се дава възможността за поглед на възникналите проблеми от различни страни.



Фиг. 3 Диаграма на Гайски

Следването на "Top-Down" методологията, също намалява количеството на промените, които се налага да бъдат направени по-късно в процеса на проектиране. Ако схемата има нужда да бъде частично променена, инфраструктурата "put in place", като част от методологията позволява промените да бъдат направени бързо и лесно. Моделите могат да бъдат променяни и качеството на системата бързо подобрявано. Инфраструктурата за симулации на така изградения дизайн е вече достъпна и може бързо да бъде приложена за проверка на направените промени.

Чип-архитекта е нова фигура в групата за дизайн. Негова задача е разработката на симулационни стратегии и план за проектирането, осигурявайки синхрон между отделните дизайнери. Главно задължение на чип-архитекта е работата на системата отговаря на очакванията след финалната реализация. Той може да бъде системен инженер който да проектира системата в блоков вид.

4. НИВА НА ПРОЕКТИРАНЕ

Чип-архитектът разработва най-високото схематично ниво на дизайна. Необходимо е първоначално да бъде уточнена системата като цяло, преди някои от съставните блокове да бъде започнат като разработка. В процеса на работа тя може да бъде променяна с цел оптимизация. Най-високото схематично ниво определя разделянето на дизайна на блокове и интерфейса между всеки един от тях. Пример за това е линията за разрешение, означена като "3V CMOS с активно ниво лог. '1'" или линията за тактуване, представена с "5V TTL преден фронт на такта". В този случай, най-високото схематично ниво на дизайна осигурява яснота по отношение на идеята която трябва да бъде реализирана.

Когато най-високото схематично ниво на дизайна е уточнено, се описват поведенчески моделите съставящи системата и работата й окончателно се проверява чрез симулационна стратегия. В процеса на разработка чип-архитекта координира всяка промяна в системата и запознава членовете на екипа с тях. Задачата на блок дизайнерите е да разработят и представят схема на транзисторно ниво (pre- и postlayout) на чип-архитекта, които я проверява чрез симулационна стратегия уточнена съобразно първоначалния план.

По време на проектирането, чип-архитектът работи с тест инженерите и разработва план за тестване и тестови програми. Достъпността на работещия системен модел в по-ранен момент от процеса на проектиране, позволява на тест инженерите да започнат разработката на тестови програми по-рано. Тази практика позволява дизайна да бъде готов за производство непосредствено след завършване на неговата разработка, което пък от своя страна намалява значително времето за излизане на крайното изделие на пазара.

Важно направление в "Top-Down" методологията, е възможността за разработката на изчерпателен план за проверки, което я определи като водеща в процеса на проектиране. Процесът започва с определянето на различни зони, специфични за дизайна. Разделянето на зони е в зависимост от това как те ще бъдат тествани. Стратегията определя как тестовете ще потвърдят работата на системата и кои блокове ще бъдат представени на транзисторно ниво, по време на тестването. Блоковете за които са се използвали теоретични модели е необходимо да бъдат идентични във всеки един от етапите на тестването. Началото на този етап започва при изготвяне на стратегията за проектиране. Тогава за всеки един от блоковете се изработват множество различни модели, целящи намирането на оптимум при описанието на системата.

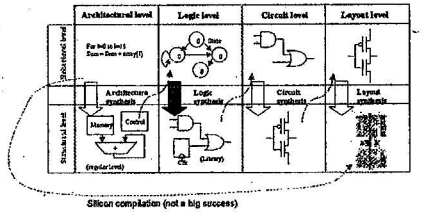
Много важно е по време на етапа на дефиниране на отделните модели, те да бъдат възможно най-опростени. Първоначално моделите съдържат базовите функции, след което към тях се добавят при необходимост допълнителни функции. Също важно е когато моделът се съставя, да се обърне внимание по-скоро на неговото поведение от колкото на неговата структура. Опростената формулировка на взаимните връзки между сигналите, работещи

като един обект се предпочита пред описанието на сложни обекти съдържащи множество сложни вътрешни структури. Това е противно на склонността на повечето дизайнери, които са добре запознати с вътрешните обработки за дадения блок и обикновено съставят точен еквивалент на блока, структурно по-сложен от необходимото.

Не е необходимо също и да се дефинира поведение на схема което е извън изискванията наложени от заданието. Това може да доведе до добавяне на допълнителен код към модела, който да го направи неподходящ за дадената ситуация и несъвместим със системата като цяло. От значение са само блокове които отговарят на поставеното задание, като входно-изходни параметри. Тези правила ще доведат до по-бързи симулации и по-малко време, необходимо за описване на моделите.

5. Следване на плавните преходи

Правилния процес на проектиране основно е резултат по-ефективни и по-изчерпателни проверки, целящи по-голяма част от грешките да бъдат откривани по-рано и след малък брой итерации на дизайна. Симулацията и стратегията за тестване са водещи при описание на системата на високо ниво, където те бързо биха могли да бъдат коригирани. Веднъж достъпни те могат да се прилагат при симулации от различни етапи на блоковете, намаляващи риска от откриване на грешки на по-късен етап от проектирането.



Фиг.4 Етапи от процеса на проектиране

Проектирането на дизайна на системно ниво, основно се извършва от системния инженер. Неговата цел е да намери алгоритъм и архитектура, които да реализират изискваната функционалност, даващи едновременно с това добри системни параметри при минимална цена. Те обикновено използват симулатори на системно ниво като Simulink или SPW, които им позволяват да изследват различни алгоритми и да променят стратегията в по-ранен етап от процеса на проектиране. Тези инструменти се предпочитат, тъй като представят дизайна като блокова диаграма, стартират бързо и имат на разположение библиотеки от предварително дефинирани блокове с общо предназначение.

Тази фаза от дизайна осигурява добро разбиране на системата на сравнително ранен етап от проектирането. Тя също позволява бърза оптимизация на алгоритмите и придвижване на основния замисъл на системата напред в процеса на проектиране, където промените биха могли да се направят бързо и не биха стрували скъпо. Нерентабилните идея отпадат рано. Симулациите са преместени също напред в процеса на проектиране, което намалява времето за симулиране, подпомага по-оптималното разделяне

на системата на блокове и финансов анализ на предявените към системата характеристики.

Когато алгоритъма бъде избран, той трябва да бъде свързан с конкретната архитектура. Това трябва да бъде изяснено от гледна точка блоковете, съставящи дизайна на системно ниво. Съобразно начина по-който те рефлектират се определя и начинът на разделяне на схемата в крайната й реализация. В блоковете трябва да бъдат представени секции от схеми, които да бъдат проектирани и проверени като възли. Интерфейсът трябва да бъде избран внимателно, така че да позволява взаимодействие между отделните блокове. Твърдо детерминиран и проектиран като носеща и обединяваща среда. Основна цел на тази фаза е коректното дефиниране на блоковете и техния интерфейс. Това контрастира в сравнение с фазата на избор на алгоритъма, където много бързо се определя изходната реакцията на разработваната схема без да се има в предвид до голяма степен последващата архитектурна реализация. Това прави MS-HDL езиците, като Verilog-AMS или VHDL-AMS предпочитани по време на тази фаза от дизайна, тъй като те позволяват коректно моделиране на интерфейса и поддръжка на симулации на различни нива.

Прехода между алгоритъм и архитектура на дизайна е връзката между двата етапа от проектирането на дизайна. Инструментите които се използват за разработка на алгоритъма са различни от тези които се използват за разработка на архитектурата. Това налага дизайна да бъде превъвеждан, което е една възможност за допускане на грешки. Това също е пречка, тестовите структури и условия разработвани по време на фазата определяща алгоритъма, да бъдат използвани при проверка на вече архитектурно реализирания дизайн.

От страна на цифровия дизайн, инструменти като SPW правят възможна реализация чрез Verilog или VHDL генерации. Като тяхна алтернатива в аналоговите или аналогово-цифровите дизайни е използването на Verilog-AMS или VHDL-AMS в алгоритмичен и архитектурен аспект.

6. Литература

1. Pratt, Gary; Jarett, Jay, ***"Top-Down*** Design ***Methods*** Bring Back The Useful Schematic Diagram", Electronic Design, 8/6/2001, Vol. 49 Issue 16, p64
2. Goering, Richard; Santarini, Michael, "Board-design twist pushes HDLs over schematic entry", Electronic Engineering Times, 1/28/2002 Issue 1203, p1
3. Elabd, Hammam; Ranganathan, Vijayaraghavan; Dholakia, Suresh, "***Top-down*** design meets ***bottom-up*** IP", Electronic Engineering Times, 03/26/2001 Issue 1159, p104
4. Pratt, Gary, "Automated ***Top-Down*** Design Rapidly Becoming Essential For Mixed-Signal Circuitry", Electronic Design, 03/05/2001, Vol. 49 Issue 5, p97
5. Child, Jeff, "Programming system combines ***top-down***, ***bottom-up*** approaches", Electronic Design, 04/20/98, Vol. 46 Issue 9, p128
6. Hubbard, Lea A.; Ottoson, Judith M., "When a ***bottom-up*** innovation meets itself as a ***top-down*** policy", Science Communication, Sep97, Vol. 19 Issue 1, p41
7. Noblitt, James S., "***Top-down*** mets ***bottom-up***" Educom Review, May/June97, Vol. 32 Issue 3, p38