

## **СВЕРХБОЛЬШИЕ ИНТЕГРАЛЬНЫЕ СХЕМЫ.**

### **ПРОБЛЕМЫ ПРОЕКТИРОВАНИЯ**

**О.В. Непомнящий, Г.А. Скотников, В.А. Хабаров, М.В. Мадудов**

*ГОУ ВПО «Сибирский федеральный университет», г. Красноярск*

**Ключевые слова и фразы:** большие интегральные схемы; высокоуровневое проектирование; микропроцессор; проектирование вычислительных систем; электроника.

**Аннотация:** Рассмотрено состояние современных проблем в области проектирования сверхбольших интегральных схем и систем на кристалле. Приведен обзор тенденций развития сверхбольших интегральных схем и намечены перспективы до 2012 г. Изложены проблемы глобального охвата проекта и межуровневых переходов при его создании. Предложены пути решения проблемы перехода между алгоритмическим описанием системы и представлением ее на вентиляционном уровне, посредством языковых методов высокоуровневого описания аппаратуры.

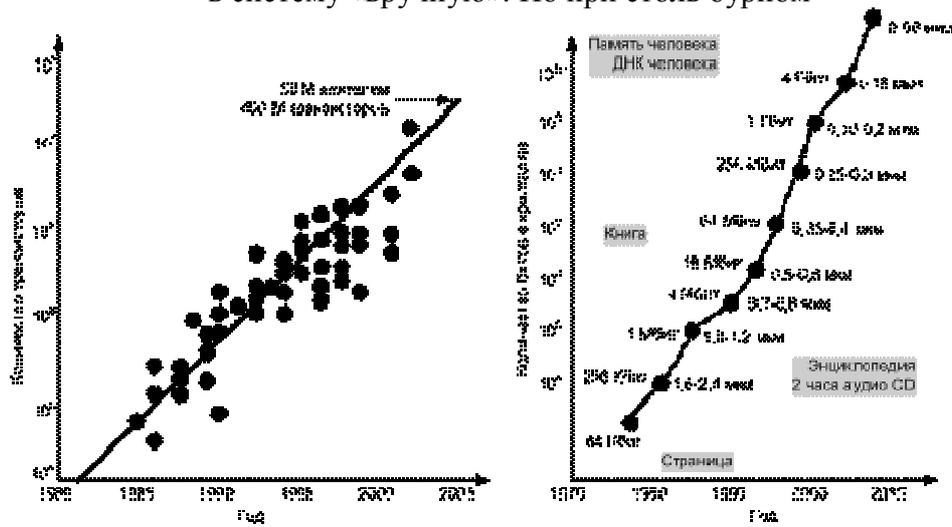
Со времени выхода первой интегральной схемы размер топологического элемента уменьшился в несколько десятков раз и сегодня приближается к диапазону 100 нм [1] (рис. 1).

Подобное уменьшение оказало существенное влияние на проектирование цифровых, интегральных схем. Особое влияние данный факт оказали на производство сверхбольших интегральных схем (СБИС). Здесь, в первую очередь, возросла сложность систем на кристалле. Параллельно, появились новые стратегии проектирования и реализации таких систем. И, наконец, субмикронные размеры ячеек выдвинули на первый план целый ряд новых проблем, которые прямо или косвенно оказывают влияние на стоимость, быстродействие, надежность, потребляемую мощность и др. аспекты функционирования СБИС.

Наиболее существенным является тот факт, что и на сегодняшний день эта тенденция полностью сохраняется. Для анализа проблем в области проектирования СБИС вернемся к истокам.

В области микропроцессорной техники рост количества транзисторов на кристалле еще более показателен (рис. 2).

Первоначально каждый транзистор в схеме вырисовывался, оптимизировался и вписывался в систему «вручную». Но при столь бурном



а б

Рис. 1. Рост степени интеграции:

а) логических ИС; б) схем памяти

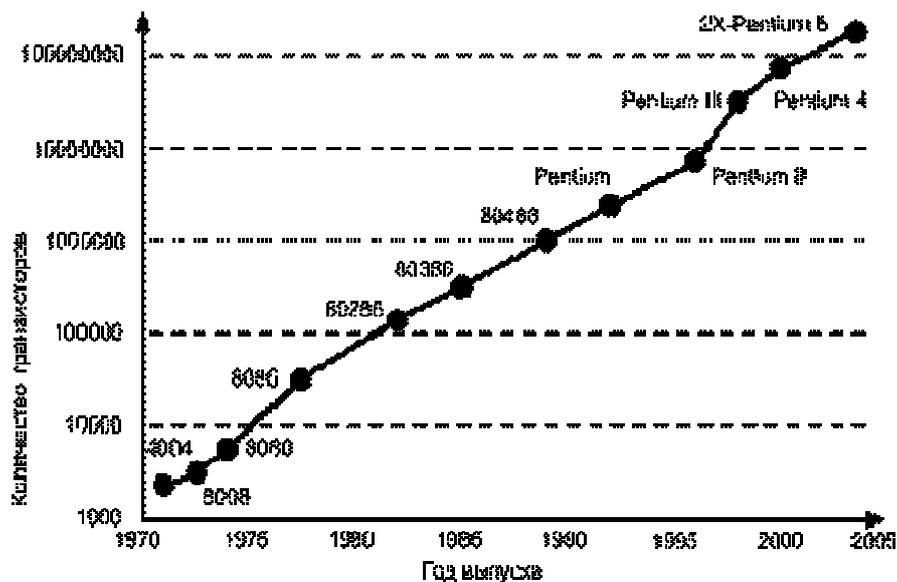


Рис. 2. Интегральная составляющая микропроцессоров Intel на начало XXI в (по данным Intel corp.)

развитии технологий основным аспектом стало время появления конечного продукта на рынке, и здесь автоматизация процесса проектирования вышла на первый план.

Как следствие столь бурного развития, методологии и стратегии создания СБИС вынуждены были оставаться жестко регламентированными (стандартизированными), поскольку именно такой подход к процессу проектирования в большей степени подвержен автоматизации.

В целях уменьшения трудозатрат, а так же повышения надежности и скорости представления проекта, стало необходимым, что бы полученные в результате проектирования

«жизнеспособные» ячейки (**IP-cores** – Intellectual Property – компоненты системы на кристалле разного уровня) как можно чаще использовались повторно.

На основании такого подхода возникло понятие иерархии проекта и Block-based design (**BBD**) – блочно-ориентированного проектирования. Разумеется, что именно такой метод оказался наиболее жизнеспособным.

Согласно [2], на одной странице можно поместить графический фрагмент типичной логической схемы состоящей из 200 вентилях, соответственно схема СБИС на 10 тысяч вентилях будет занимать порядка 50 страниц, а о схему в 500 тысяч или миллион вентилях принципиально невозможно охватить.

Таким образом, можно с уверенностью констатировать тот факт, что на сегодняшний день, одной из основных, (если не доминирующей) проблемой проектирования СБИС является глобальная проблема «охвата проекта».

Решение данной, и связанных с ней, проблем лежит в области альтернативного представления и схемотехнического описания. Здесь важно следующее:

1. Требуется модернизация традиционного BBD и модульно-иерархического подхода к описанию проекта, в частности следует четко разграничить уровни абстрагирования будущей модели проекта, с выделением строго регламентируемых межуровневых интерфейсов (рис. 3).

2. Требуется в максимальной степени исключить из цикла проектирования, все «ручные» операции, не только операции трассировки при переходе с уровня на уровень, но и операции внутрисхемной оптимизации на заключительных этапах.

3. Требуется графическое (схемотехническое) и языковое описание цифровых и аналоговых модулей с максимальной точностью учитывающее не только электрические и алгоритмические параметры описываемых модулей, но и как следствие, топологию их расположения на кристалле.

Таким образом, на сегодняшний день, СБИС следует рассматривать как сложный многокомпонентный объект. Причем здесь требуется выделить не только логические функции по преобразованию информации и обработке данных, структурную организацию многокомпонентной системы, но и учесть интегральную многослойную технологию полупроводникового прибора.

Традиционно СБИС рассматривают согласно трем уровням представлений [3]:

- поведенческое (функциональное) представление;
  - структура СБИС;
  - топология (геометрия) системы.
- Причем каждый уровень представления разбивается на дополнительные подуровни. Например, в структуре СБИС можно выделить:
- уровень структурных модулей. Процессор, память, готовый модуль и др. (Processor, Memory, Switch — PMS-level);
  - регистровый уровень. Здесь задаются межрегистровые связи и описываются мелкие функциональные блоки. (Register Transfers Level — RTL-level);
  - уровень логических вентилях (Gates, Gate level);
  - схемотехнический (транзисторный) уровень (Transistor level).

Процесс проектирования представляет собой последовательный спуск по означенным уровням с одновременным перемещением по оси иерархии других областей (рис. 3). Таким образом, поуровневый характер проектирования внутри каждой области сочетается с согласованным движением по осям других измерений (областей) [2].



4. При высокой сложности современных проектов очень сложно провести анализ различных вариантов проекта. Здесь практически отсутствует вероятность выработки оптимальной системы. Это связано как с временными затратами, так и требуемым увеличением вычислительных мощностей.

5. Такой проект очень сложно модифицировать. Легко подвергаются модификации только уровень MatLab, а при переносе модификаций на RTL – уровень вновь встают все вышеозначенные проблемы.

6. Существует проблема представления кода RTL – уровня, поскольку для различных СБИС коды реализации могут отличаться. Например, код реализации для FPGA может отличаться от кода для ASIC.

7. Решение означенных проблем возможно добавлением промежуточного представления проекта на доступном обоим уровням языке представления, поскольку возникают они, именно из-за отличий высокоуровневого представления между MatLab и RTL описанием.

Такое промежуточное представление должно базироваться на языке C/C++. Это обусловлено следующими положениями:

1. Концептуально языки MatLab и C – очень схожи между собой, поэтому даже ручной или частично автоматизированный переход между ними не вызовет серьезных затруднений. Таким образом, устраняются различия в представлении проекта между алгоритмистами и инженерами.

2. Верификация C – модулей проекта является несложной и достаточно быстросействующей операцией. Здесь можно проводить распараллеливание процессов по модулям, организацию очередей и др. например средствами MATLAB Simulink.

3. Средствами автоматизированного проектирования C – проект достаточно просто преобразуется в RTL – описание, т.е. имеется актуальная возможность генерации RTL – кода на основе C – представления проекта.

4. Возрастает простота и скорость верификации, поскольку представления на MatLab и C намного более компактны описания на RTL – уровне. При этом, чем сложнее проект тем существенней будет разница в скорости.

5. Открываются перспективы для исключения RTL – представления из цикла проектирования. Поскольку возможна генерация проекта на вентильном уровне непосредственно из C – описания.

6. При компактном C – представлении можно с небольшими затратами провести апробацию нескольких вариантов проекта. Таким образом, вероятность нахождения наиболее оптимального варианта существенно увеличивается.

7. Достаточно просто вносить изменения в проект, поскольку на C или MatLab уровне модификация проста и доступна, а RTL – уровень генерируется автоматически из C – представления.

8. Полностью отсутствует привязка к реализации. Поскольку C – представление абсолютно не зависит от способа воплощения кода и может быть транслировано как для FPGA так и для ASIC.

Таким образом, предложенное решение должно основываться на применении C – языков описания аппаратуры, например System-C или Handel-C, хорошей альтернативой которым является Catapult-C.

Результатом такого подхода, помимо вышеперечисленного, будет является высокая скорость трансляции и моделирования (System-C), или возможность получения вентильного описания при помощи специализированного пакета программ, например DK Design Suite компании Celoxica (Handel-C), а также практически полное соответствие исходному C – коду (Catapult-C). Более того при проектировании в Catapult-C имеется возможность задавать микроархитектуру системы еще на этапе формального описания.

Тем не менее, применение известных программных средств создания и верификации проекта, уже в ближайшем будущем потребует очередного «скачка». Здесь намечаются

прорывы в области применения языковых средств не в «чистом» виде, а посредством графического описания систем с автоматической генерацией результирующего кода.

В рамках данного развития, известные программные продукты, должны быть так же переориентированы на учет топологии и расположения элементов на кристалле, начиная уже с перехода на RTL – уровень или минуя его.

*Список литературы:*

1. Рабаи, Жан М. Цифровые интегральные схемы, методика проектирования / Жан М. Рабаи, Ананта Чандракасан, Боривож Николитч. Изд. 2. Москва. С-Петербург. Киев., Вильямс, 2007. – 894 с.
2. Суворова, Е.А. Проектирование цифровых систем на VHDL / Е.А. Суворова, Ю.Е. Шейнин. СПб. : БХВ-Петербург, 2003. – 576 с.
3. Parnell, K. Programmable Logic Design Quik Start Hand Book / K. Parnell, N. Mentha. Xilinx Corp., 2002. – 201 p.
4. Kang, S. CMOS Digital Integrated Circuits. Analysis and Design / S. Kang, Y. Lebelevici. Boston, McGraw-Hill, 1999.
5. Селиванов, И. Маршруты высокоуровневого синтеза / И. Селиванов // Современная электроника. – М. – 2007 г. – №6. – С. 38–42.

**Very Large Scale Integration. Designing Problems**

**O.V. Nepomnyashchiy, G.A. Skotnikov, V.A. Khabarov, M.V. Madulov**

*Sibir Federal University, Krasnoyarsk*

**Key words and phrases:** computer systems designing; large integral schemes; electronics; microprocessor; high-level designing.

**Abstract:** The paper studies the present-day problems in the area of designing very large scale integration and crystal systems. The review of trends in the development of VLSI is given; the prospects up to 2012 are outlined. The global problems of the project and inter-level jumps in its creation are presented. The ways of solving the problem of transition between algorithmic description of the system and its presentation on switch level through language techniques of high level description of equipment are proposed.