АНАЛИЗ УРОВНЕЙ ПРЕДСТАВЛЕНИЯ ПРИ ПРОЕКТИРОВАНИИ СИСТЕМ НА КРИСТАЛЛЕ

Автор: Сергей ГРИЦКОВ
Научный руководитель: д.т.н., доцент Геннадий БОДЯН

Технический Университет Молдовы

Abstract: *the design of systems on a chip at different levels of abstraction - the behavioral, structural and physical are presented in this article. The advantages and disadvantages of each level of abstraction are analyzed on an example of designing of the eight bit register on the CPLD in the system of simulation QUARTUS 9.1. Conclusions about the choice of appropriate levels of abstraction are made.*

Ключевые слова: *системы на кристалле, проектирование, уровни представления, анализ, описание.*

Проектирование систем на кристалле является универсальной и многоплановой дисциплиной, объединяющей в себе методы проектирования законченных аппаратно-программных комплексов, встраиваемых систем на основе стандартных процессоров и процессорных ядер, разработки встроенного программного обеспечения, программируемых ПЛИС, полузаказных и заказных интегральных схем.

В общем случае, система на кристалле может включать в себя различные типы блоков: программируемые процессорные ядра, блоки ASIC, блоки программируемой логики и другие.

Большое распространение получили полузаказные БИС на основе матричных структур. Они представляют собой предварительно размещенные, но не соединенные базовые логические элементы, расположенные в виде матрицы. К таким БИС относятся базовые матричные кристаллы, масочные и лазерно-программируемые ПЛИС (MPGA, LPGA), а также перепрограммируемые структуры ПЛИС (FPGA).

Проекты систем на кристалле могут быть представлены на различных уровнях. Уровни представления проекта отличаются по типу отображаемой информации, и могут быть классифицированы в виде трех типов: поведенческий, структурный и физический.

В ***поведенческом*** представлении описано только функциональное поведение системы, и проект представляется как «черный ящик», имеющий зависимость выходного сигнала от входного. Поведенческое описание задает алгоритм, реализуемый системой.

Структурное представление детализирует проект, вводя информацию относительно компонентов системы и их взаимодействия.

Детальные физические характеристики компонентов определены в ***физическом*** представлении, включающем информацию о размещении и трассировке.

Как следует из анализа состояния проектирования на различных уровнях абстракции, в 1990 году реализация проекта на физическом уровне занимала 90% всего объёма проектных работ, в 2000 году эта доля сократилась до 55% и к 2010 году проектирование на структурном и функциональном уровнях составило 70% общего объёма работ.

Для проектирования систем на кристалле современными разработчиками используются языки **VHDL**, **Verilog** и **SystemC**. Первые два языка были разработаны в середине 80-х и до сих пор являются наиболее распространенными классическими HDL (Hardware Description Language)-языками; язык SystemC – является современной разработкой, находящейся в стадии развития.

На рис.1 представлен результат сравнения языков VHDL и Verilog при проектировании на различных уровнях абстракции.

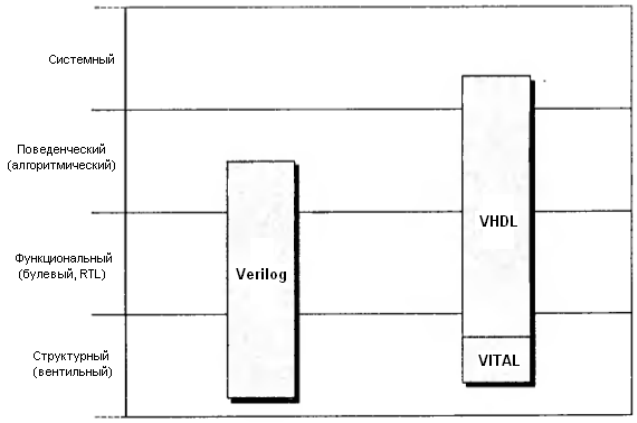

Рис.1. Пример сравнения языков VHDL и Verilog.

Главным вопросом при проектировании является выбор уровня описания, который позволит получить наилучшие результаты при проектировании. Для этого необходимо проанализировать преимущества и недостатки каждого уровня описания систем на кристалле.

Рассмотрим более подробно каждый уровень описания в отдельности на примере проектирования восьми битного асинхронного однотактного регистра на основе D-триггеров на ПЛИС в среде QUARTUS 9.1.

Представление проекта на поведенческом уровне

Листинг программы проекта приведен ниже:

```
Begin                          -- начало описания проекта
 process (Reset ,D)            -- объявление параллельного оператора
  begin
   if Reset='0' then           -- если на входе Reset – “0”, то
     Q <= "00000000";          -- все восемь линий выходов сбрасываются в “0”,
    else                       -- иначе
     Q <= D;                   -- на выходах Q появляются данные с о входов D
   end if;
 end process;                  -- завершение описания проекта
```

В листинге программы “Reset” – вход сброса, “D” – вход данных и “Q” – выход данных.

На рис.2 приведены результаты проектирования:

Revision Name	Register_8
Top-level Entity Name	Register_8
Family	MAX II
Device	EPM240F100C4
Timing Models	Final
Met timing requirements	Yes
Total logic elements	8 / 240 (3 %)
Total pins	17 / 80 (21 %)

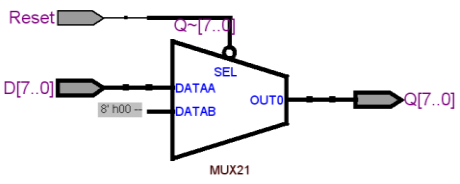

Рис.2. Отчет компилятора-синтетезатора на основе выполненного проекта (слева) и схема устройства, полученная при компиляции проекта (справа).

При проектировании был реализован регистр на 8 логических ячейках, но на схеме вместо регистра получен мультиплексор, который выполняет требования задания.

Вывод: листинг программы достаточно краток и понятен, но невозможно определить заранее какие элементы будут задействованы для реализации проекта, что влияет на быстродействие устройства, т.к. оно может быть реализовано на гораздо большем количестве элементов, чем предполагалось. Это, в свою очередь, исключает какую-либо возможность предсказания временных задержек до компиляции всего проекта.

Представление проекта на функциональном уровне

Листинг программы проекта (основное тело программы) приведен ниже:

```
Begin                                   -- начало описания проекта
 Q(7) <= (D(7) and Reset);              -- выходу Q7 назначается сигнал со входа D7, при учете
Reset
 Q(6) <= (D(6) and Reset);              -- выходу Q6 назначается сигнал со входа D6, при учете
Reset
        .       .       .
 Q(0) <= (D(0) and Reset);              -- выходу Q0 назначается сигнал со входа D0, при учете
Reset
```

На рис.3 приведены результаты проектирования:

Flow Status	Successful - Wed Nov 03 22:25:55 2010
Quartus II Version	9.1 Build 222 10/21/2009 SJ Full Version
Revision Name	Register_8
Top-level Entity Name	Register_8
Family	MAX II
Device	EPM240F100C4
Timing Models	Final
Met timing requirements	Yes
Total logic elements	8 / 240 (3 %)
Total pins	17 / 80 (21 %)
Total virtual pins	0
UFM blocks	0 / 1 (0 %)

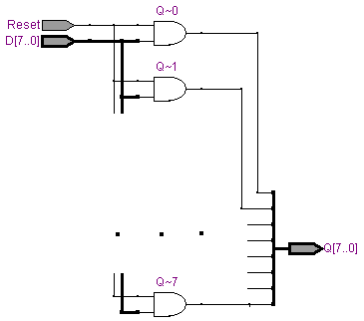

Рис.3. Отчет компилятора-синтетезатора на основе выполненного проекта (слева) и схема устройства, полученная при компиляции проекта (справа).

При проектировании был реализован регистр на 8 логических ячейках, но на схеме вместо регистра получена комбинационная схема, которая выполняет необходимые условия в соответствии с требованиями задания.

Вывод: листинг программы более сложный, чем при поведенческом описании (сложнее определить, что проектируется именно регистр), реализовать проект удалось лишь на логических вентилях “2И”, которые в результате и использованы на схеме. При сравнении с поведенческим описание на вентильном уровне сложнее, однако, здесь заранее известен каждый элемент, используемый в схеме, что позволяет еще на стадии проектирования определить все временные задержки прохождения сигналов.

Представление проекта на функциональном уровне

Листинг программы проекта приведен ниже:

```
COMPONENT D_Trigger IS                          -- описание компонента
    PORT                                        -- D_Trigger
        (                                       -- который содержит
        Reset   : in std_logic;                 -- вход Reset
        D       : in std_logic;                 -- вход D
        Q       : out std_logic                 -- выход Q
    );
```

```
  END COMPONENT;
Begin                                                    -- начало описания проекта
        D7 : D_Trigger                                   -- подключение
D_Trigger
        Port map (Reset=>Reset, D=>D(7), Q=>Q(7));       -- задаем значения его
линий
        D6 : D_Trigger                                   -- подключение
D_Trigger
        Port map (Reset=>Reset, D=>D(6), Q=>Q(6));       -- задаем значения его
линий
                .       .       .
        Port map (Reset=>Reset, D=>D(0), Q=>Q(0));
end Register_8_Architecture;                             -- завершение описания
проекта
```

На рис.4 приведены результаты проектирования:

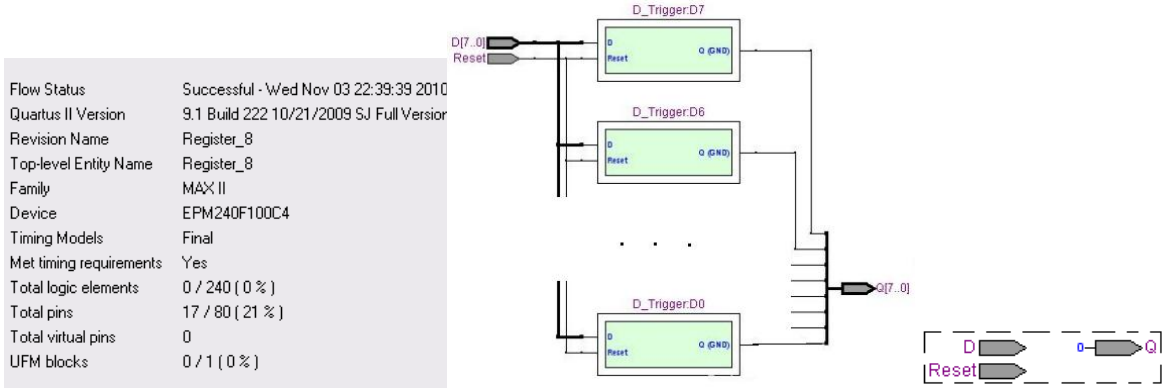

Рис.4. Отчет компилятора-синтетезатора (слева), схема устройства, полученная при компиляции проекта (в центре) и содержимое ячеек D_Trigger:D (справа).

Вывод: При реализации проекта не было использовано ни одной логической ячейки, но на схеме присутствует регистр, состоящий из восьми триггеров, что и требовалось получить изначально. Не использованы логические ячейки ПЛИС, так как триггеры ничего не выполняют. Из этого можно сделать заключение, что на функциональном уровне можно лишь представить проект в виде иерархических блоков, что удобно при изучении проекта, но функциональный уровень должен дополняться описаниями на поведенческом либо структурном уровнях.

Библиография

1. Бибило П.Н. *Основы языка VHDL. СОЛОН-Р*, Москва, 2002.
2. Суворова Е. *Проектирование цифровых систем на VHDL*. БХВ-Птербург, Санкт-Петербург, 2003.
3. Максфилд К. *Проектирование на Плис. Курс молодого бойца.* Додэка-XXI, Москва, 2007.