

Языки описания аппаратуры (HDL-языки)

разновидности

- языки низкого уровня (аналоги языков программирования типа ассемблера)
- высокого уровня

- ***Языки низкого уровня*** ближе к аппаратным средствам, вследствие чего представляют для компиляторов потенциальные возможности создания проектов с более выигрышными параметрами. Имеют жесткую ориентацию на определенную аппаратуру и производящую ее фирму.
- Примерами таких языков могут служить языки PLDASM (фирма Intel), AHDL (Фирма Altera) и ABEL (Фирма Zilinx).

- Языки высокого уровня менее связаны с аппаратными платформами и поэтому более универсальны. Среди них наиболее распространены языки VHDL и Verilog.

Язык VHDL

- Язык VHDL появился в начале 80-х годов по заказу организации Министерства обороны США. Первая его версия, предназначенная в основном для унификации описаний проектов в различных ведомствах, была принята в 1985 году.

- Синтаксические конструкции языка содержат две составляющие – общеалгоритмическую и проблемно-ориентированную

- Общеалгоритмическая составляющая языка достаточно традиционна и содержит как традиционные операторы действия (присваивание), условия (IF), выбора (CASE), цикла (LOOP), вызова процедуры), так и традиционные типы данных: числовые, логические, символьные, перечислительные и агрегатированные (массивы, записи, файлы).

- Каждый объект проекта состоит, как минимум, из двух различных типов описаний: описания интерфейса и одного или более архитектурных тел. Интерфейс описывается в объявлении объекта (entity declaration) и определяет только входы и выходы объекта.

- Для описания поведения объекта или его структуры служит архитектурное тело (architecture body). Чтобы задать, какие объекты использованы для создания полного проекта, используется объявление конфигурации (configuration declaration).

- Язык VHDL является очень распространенным языком описания аппаратуры.

Язык Verilog

- Язык Verilog был разработан фирмой Gateway Design Automaton как внутренний язык симуляции. Cadence приобрела Gateway в 1989 г. и открыла Verilog для общественного использования. В 1995 г. был определен стандарт языка — Verilog LRM (Language Reference Manual), IEEE1364-1995. Таким образом, датой появления языка Verilog следует считать 1995 г.

- Основной структурной единицей Verilog описания является `module`. Модуль соответствует `entity` в VHDL. Модуль описывается ключевыми словами `module` — `endmodule`. В файле может быть описано несколько модулей. Другие модули могут подключаться к цепям модуля, образуя иерархическую структуру. При запуске Verilog симулятор строит иерархическое дерево из всех модулей, которые обнаружены в файлах, поданных на вход симулятора, и находит модуль верхнего уровня. Если таких модулей несколько, то происходит ошибка.

- Как правило, модуль содержит список портов — интерфейсных сигналов, которые служат для подключения его в других модулях. Порты бывают трех типов input — входы, output — выходы, inout — двунаправленные. Входы и двунаправленные порты должны иметь тип wire, а выходы могут быть как wire, так и reg. Построение иерархии (подключение модулей) возможно двумя способами: по имени (указываются имена портов, использованные при описании модуля) или по расположению (порядок сигналов такой же, как в описании модуля)

Язык SystemC

- Необходимость создания нового языка описания аппаратуры возникает в том случае, если существующие языки не способны решать возникающие задачи. С одной из трудноразрешимых задач столкнулись самые популярные и распространенные языки VHDL и Verilog – это постоянно растущая сложность разрабатываемых проектов и необходимость в универсальном способе описания, пригодного для всех уровней проектирования и независимого от формы реализации проектов. Попыткой решить возникшие трудности стало использование языка SystemC.

- В 1999 году вышла в свет первая версия библиотеки SystemC 0.9.

- SystemC представляет собой надстройку стандартного языка программирования C++, реализованную в виде отдельных библиотек специальных классов. Данные библиотеки содержат в себе конструкции, позволяющие создавать эффективные и точные модели программных алгоритмов, аппаратных архитектур, интерфейсов и схем на системном уровне, т.е. практически всех компонентов встроенных систем. Такой подход имеет значительный потенциал, так как основан на однородном описании C++ и легко позволяет моделировать, тестировать системы, рассматривать альтернативные архитектуры. Кроме того, команде проектировщиков может быть предложено развернутое описание процесса работы всей системы. Это описание представляет