An XACT science

Building an IP-XACT design and verification environment with DesignWare IP. By **John Swanson**.

With more IP components and growing time to market pressures, designers are looking for a way to build and update SoC designs easily. IEEE1685 (IP-XACT) was designed to fit this requirement and Synopsys' DesignWare digital IP cores support this standard, with the IP-XACT file generated as a view of the configured core or subsystem.

As more tools support the standard, it will be used in a variety of ways, ranging from building the documentation chapters of a configured component (such as register descriptions) to batch flows that can regenerate the RTL for a component, subsystem or SoC. Companies will also use it to extract the information they need to build a verification environment.

What is IP-XACT?

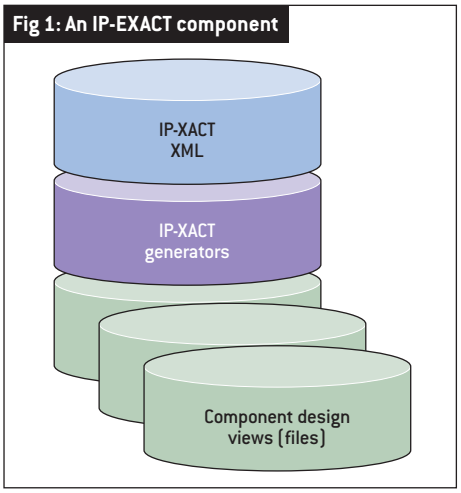
IP-XACT is an XML schema for language and vendor neutral IP descriptions that includes a generator interface for 'plug in' functionality. It is design language neutral, design tool neutral, efficient and proven.

The schema is built on the XML (W3C) standard, with a standardised API for generator integration called the Tight Generator Interface. The schema, first released by the SPIRIT Consortium, is now an IEEE standard that is validated and released in accordance with the IEEE policies published as IEEE-1685.

XML schemas are used to define the legal building blocks of an XML document or document structure. An XML schema defines:

- elements and attributes that can appear in a document
- which elements are child elements
- the number and order of child elements
- whether an element is empty or can include text
- data types for elements and attributes
- default and fixed values for elements and attributes

Fig 1: An IP-EXACT component



It is important to note that XML does not do anything! XML was created to structure, store and transport information and is just plain text. Software that can handle plain text can also handle XML. XML schemas can also be processed by XML aware applications, which use the XML tags specifically to do different things. For example, the IP-XACT schema can be used to drive the assembly of components.

The functional meaning of the tags depends on the nature of the application. With XML, you invent your own tags, as XML has no predefined tags. XML is designed to support schemas like IP-XACT, which was designed for the design and verification of IP based SoCs.

IP-XACT provides XML descriptions of components and systems. A component has several attributes that can map directly to XML.

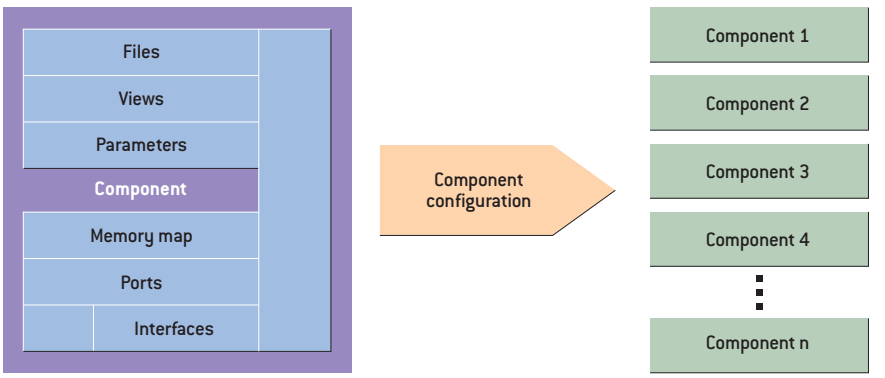
These include:

- memory maps
- bus interfaces
- views (additional data files)
- file sets
- registers
- ports
- parameters
- generators

When multiple components are connected, it becomes an IP-XACT design file. This includes the attributes above, as well as the interconnect information of all the components in the design. IP-XACT files and generators can be thought of as simply 'another view' of an IP block (see fig 1).

Synopsys provides digital IP which users can integrate into their designs and then configure for the target application. Synopsys generates the IP-XACT file based on the users' configuration. This allows designers to have an XML master

Fig 2: Component configuration

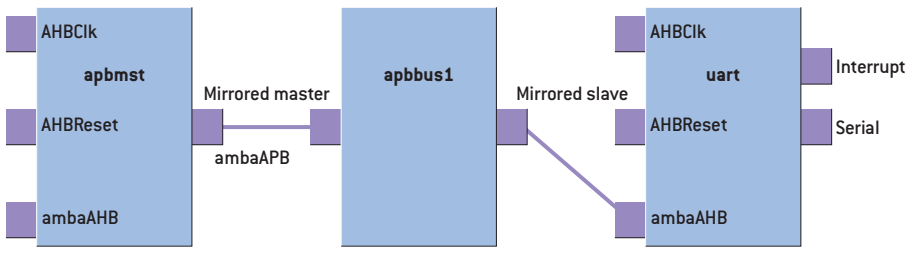


view of their configuration without having to deal with IP-XACT limitations related to configurable IP. Fig 2 shows how one component can be configured to generate many different components, each of which has its own IP-XACT representation.

A mature IP-XACT flow does not require knowledge of the XML schema any more than an internet browser requires knowledge of XML; it is metadata that sits behind the tool.

IP-XACT allows components with associated interfaces to be connected automatically. Consider the simple example of an AMBA APB master connecting through an APB bus to an APB slave. As the APB interface has been defined, the three components can connect automatically. An example of this is shown in fig 3.

Fig 3: A simple APB master to slave connection using the IP-XACT interface



Companies using IP-XACT as part of their design and verification have noted how using the standard not only has low adoption costs, but also provides the data needed to expand to verification, software tool support and documentation.

Building IP-XACT digital IP

There are multiple ways in which IP-XACT digital IP can be built. At the centre of Synopsys' solution is coreBuilder. This is an IP packaging tool that allows you to generate an IP-XACT view or a coreKit (which includes features not supported by the standard). Essentially, coreKits are a knowledge base that is configured with the coreConsultant tool.

The coreBuilder tool allows the designer's knowledge to be captured in the IP package and supports both GUI and command line use. When coreBuilder is started, the user is presented with a dialogue box that allows them either to select a coreKit (default) or an IP-XACT component. After making that selection, the user is then presented with the development window.

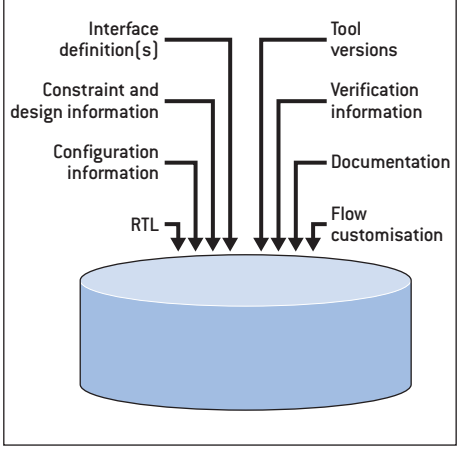
The development window walks users through the flow of adding all of the information related to the IP. Typical data includes:

- source code
- configurability information
- design constraints
- documentation
- testbench
- customisations
- assembly intent / interfaces
- tool versions supported
- IP specific information
- different views (C/behavioural/gate) for the IP component.

The minimum data required to build a coreKit or IP-XACT component is the source code: all other information is optional and can be added incrementally.

At any point when using the GUI mode, the user can save a batch file to return to that point in development. Designers often choose to do the initial IP packaging with the GUI and then write out the batch file and use it for updates to the core. The batch file is a TCL script that can be edited as updates are made to the IP.

Fig 4: Component info. included in coreBuilder



Once the information collection process has been completed, coreBuilder generates a coreKit or IP-XACT component. This will include all of the information provided, including help files and configuration dialogues (see fig 4).

How IP-XACT is generated

Once the configuration information, registers, memory maps and interfaces are included in the coreKit or IP-XACT component,the IP-XACT component or design file can be recreated at any time using coreConsultant. This will guide users through the process of configuring and optionally implementing the design on an fpga for prototyping or targeting a specific library for SoC implementation.

While coreConsultant generates coreKits for a single IP core, coreAssembler can be used for interface and rules based assembly of coreKits, IP-XACT components or RTL source code for multiple IP cores. It can also be used to add custom tasks, allowing it to work in different design and verification environments. In addition to support for importing an IP-XACT component, core Assembler generates connectivity reports, register reports and more.

Users can get a head start in developing a verification environment using the information available in the generated IP-XACT component. To illustrate this, let's examine a portion of the schema from a simple design: a UART with an APB interface.

Users will want to know the registers and how to connect a Verification IP (VIP) component. Using the various features available, the IP-XACT XML can be parsed to build and configure a testbench. With this information formatted for easy parsing, a data model can be built to enable the assembly of a testbench, configuration of a VIP component and, in some cases, set up some stimuli to execute on the testbench.

As Synopsys' VIP supports the AMBA protocol, coreAssembler can generate a connectivity test that will build a testbench, configure the VIP, and execute a write/'read expect' test to the registers of the components in your design – an excellent starting point for the larger SoC verification process.

Author profile:

John Swanson, is senior marketing manager, DesignWare IP, with Synopsys.