Simplified FPGA Design with Robei

Guosheng Wu
Robei LLC, Henderson, NV, USA
robei@robei.com

Abstract – Robei is a tiny cross platform FPGA design tool that aims to simplify design procedure, transparent intellectual properties and reduce complexity. It makes FPGA design like playing with boxes by breaking down hardware into three basic elements: module, port and wire. Through these elements, engineer can implement either top-down or bottom-up design. Standard Verilog code can be integrated with other EDA tools generated from design diagram. Robei also runs on embedded platforms, which makes it distinctive from other EDA design software.

Keywords: Robei; FPGA; Verilog; EDA; Simulation;

1 Introduction

FPGA is a technology intensive field full of innovations. Because of its low cost, easy to change and short time to market advantages, plenty of companies and individual engineers choose it to prototype products. Hardware engineers are focusing on developing their new designs in programming languages like VHDL and Verilog [1].

Although the future is very bright, there are still some obstacles that prevent FPGA market from fast growing, like high background requirement, opaque intellectual properties, and huge, complexity design tools. First of all, FPGA design requires not only knowledge of physics and circuits, but also digital design and logic synthesis. However, most of such knowledge is offered in universities and graduate schools, which limits the market growing. Second, FPGA tools are unique to each other. In order to implement a project on certain FPGA chips, the designer must stick to FPGA vendor’s software and intellectual properties. Design engineers need to spend a lot of time to get familiar with these tools for the first time. Third, FPGA tools are huge in size and complex in contents. Current size of FPGA design software already counts in gigabytes. Bugs are increasing as the software size increase, which lead to the phenomenon that most engineers choose to use older version of design tools. Because there are so many options and features in design tools, which already beyond engineers' play-to-learn ability, so FPGA vendors need to spend a lot of time and money on training customers.

On the other hand, the successful stories of Apple’s iOS and Google’s Android [6] on mobile platforms proved that user interface become more and more important for customers. A lot of software companies already transferred their products from personal computer to mobile platform. However, FPGA design on mobile platform is still a challenge due to the high complexity and huge size of FPGA design tools.

Fig.1 Robei user interface

This paper proposes a cross platform FPGA design software named Robei, shows in Fig.1. It is based on the most popular cross platform GUI framework, QT [7] which is already ported to iOS and Android platform. Robei can be re-compiled on many platforms without much modification. It aims to simplify user interface for FPGA design, transparent intellectual properties and reduce design complexity. The modern user interface of it combines diagram design method to represent circuit connections and coding method for algorithm inputs. Robei is designed to be as simple as possible for engineers. Let them mange in 15 minutes as long as they are familiar with Verilog language. Any pre-designed and system provided models are transparent to users. Property editor offers the most convenient method for viewing and modifying properties for each element. There are only three elements to representing the circuits: module, port and wire. By simply playing with these elements, FPGA designers can construct project by either bottom-up or top-down mechanism easily.

The remainder of this paper is organized as follows: element description is illustrated in Section 2; Code
generation is presented in Section 3 and a simple FIR filter design example is depicted in Section 4; screen captures of Robei running on Android platform is shown in section 5; conclusions and future work are given in Section 6.

2 Robei Elements

Robei employs three elements to represent Verilog components in hardware design: module, port and wire. In Verilog, circuits are represented by a set of "modules". A module may be only a gate, a flip-flop, a register, but also can be an ALU, a controller or a SOC system. We can consider a module as an abstract chip, which have different ports (pins) to communicate with other chips. A finished design module can be considered as model, which locates in "Toolbox" area and can be reused. Port is the interface channel for each module and model. Wire is used to connect ports on different module or model for signal transmission.

2.1 Module

In Robei, a module, the basic in a design, can be considered as a black box. Inside this box, designer can place ports algorithm codes and models. Each module can have zero or more ports for communication with other modules or models. The code view tab allows engineers to add or modify algorithm code easily to realize certain behaviors.

Fig. 2 module and model: only the biggest rectangle (coder_example) is module, the other smaller rectangles and rectangles in "ToolBox" are all models, which are pre-designed modules.

Based on usage status, a module can have different types. Currently under developing one has the type of "module", but once it is used in other modules, the type will automatically change to "model" as only certain properties can be modified in order to keep consistency with previous design. Fig.2 illustrates "module" and "model". There is another special type named "testbench", which is the top level design with stimulate code for simulation.

2.2 Port

A port may correspond to a pin on a chip, an edge connector on a board, or any logical channel of communication with a block of hardware. The detail properties are listed in fig.3. The type of port varies a lot as Robei supports many types in Verilog, like reg, wire, tri, supply, etc. There is "Datasize" option for port, which specifies the size of port if it is a bus. Some interesting features worth to mention are port can only slide on edges of module, and when module moves, port keeps sticking to edges all the time.

![Port properties](image)

Fig. 3 Port properties

2.3 Wire

Wire connects two ports and responses for signal transmission. Most of time, wire will inherent the color and data size from the first connected ports (as shown in fig.4). Based on different data size, wire has different thickness. Robei helps to check whether two connected ports have same data size or not when simulation start.

![Wire](image)

Fig. 4 Wire

Wire connects two ports and responses for signal transmission. Most of time, wire will inherent the color and data size from the first connected ports (as shown in fig.4). Based on different data size, wire has different thickness. Robei helps to check whether two connected ports have same data size or not when simulation start.

Module, port and wire are basic elements and used very often. Robei break down FPGA structure level design to these simple elements so even high school student can play with it.
3 Code Generation

The simplified user interface helps designer to reduce the code input and avoid mistakes. Instead of typing complete code for a project, hardware designers just need to write their core algorithm with code editor, while the interface design can be completed by playing with elements. For example, fig. 5 and fig. 6 show a simple counter design and its core algorithm that requires user to type in. The code under "Code start here" are core algorithm part. The other part like parameters, port declaration, module instantiation and extra signal declaration are generated based on designed property by Robei.

```
3
Code Generation
The simplified user interface helps designer to reduce the code input and avoid mistakes. Instead of typing complete code for a project, hardware designers just need to write their core algorithm with code editor, while the interface design can be completed by playing with elements. For example, fig. 5 and fig.6 show a simple counter design and its core algorithm that requires user to type in. The code under "Code start here" are core algorithm part. The other part like parameters, port declaration, module instantiation and extra signal declaration are generated based on designed property by Robei.

Fig. 5 Simple counter example
```

```
```
relation between \( f[n] \) and coefficients shows in equation 3 when \( n > L - 1 \).

\[
f[n] = \sum_{k=0}^{L} h[k] \times u[n] = \sum_{k=0}^{L} h[k]
\]  

(3)

The output signal value proves statement above in simulation result (fig.8).

![Fig.8 Step response of FIR filter](image)

5 Embedded Platform

Thanks to Necessitas, which is the code name for porting of Qt to Android Operating System, and Verilog Behavioral Simulator (VBS) which is the first cross platform Verilog simulator designed by Jimen Ching, Robei is the first FPGA design tool that can perform simulation on embedded platforms like Android. Here are some images captured on android 2.3 platform when running. With Robei, developers can realize their design anywhere on mobile phones or tablets.

![Waveform simulation](image)

6 Conclusions

Robei starts a brand new way for FPGA design by providing simplified hardware design procedures, transparent property and incredible mobility. The goal of this design tool is letting everyone to play with FPGA design at anywhere and increasing the innovations from new sources other than hardware engineers.

The design concept of Robei is not only suitable for FPGA design, but also feasible in other fields. In future, it will be improved and extended for other research and industrial field to provide more practical values.

7 References


