AN1566
Application Note

In System Prototyping Using HDLs and FPGAs

Prepared by
Thomas G. Felske
Doug Hergatt
Motorola Programmable Logic Group

3/96

REV 0

MOTOROLA

In System Prototyping Using HDLs and FPGAs

Introduction

This paper describes the rapid prototyping method used in the development of the power control logic for a complex communication system. VHDL and SRAM based FPGAs were used to optimize the HDL code before it was merged into an existing ASIC. This method will prove an effective way to verify the HDL code's functionality while reducing development time.

Application

A large state machine needed to be developed and verified to monitor and control the receiving and transmitting function of a commercial space communication system. The underlying goal of the design was to minimize overall system power consumption by intelligently managing the sequence (state events) of transmit, receive, and power up/down steps inherent to the system. Long system level simulation runs, typically two days per simulation would be required to verify an HDL modification. The use of an FPGA as an in system verification vehicle to model new HDL code gave instant real time feedback of the design performance.

Design Environment

To develop the large scale communication system, a high level design environment for signal processing was chosen. The Signal Processing Worksystem (SPW) was used as the behavioral front end model for all logic in the system. The VHDL derived from the SPW tool was then the input to Synopsys for logic synthesis and technology mapping to the ASIC and FPGA architectures. For the FPGA, the Motorola SRAM based MPA1036 was used. The Motorola FPGA place and route tools provided the bitstream to program an EPROM with the FPGA configuration. System level verification was completed in a test environment as shown in Figure 1, *FPGA Prototyping Application*.

The system evaluation was controlled by the System Test Environment (STE) software that was resident on an HP workstation. The System Test Environment could exercise the desired receive and transmit function, while logic analyzers and current probes monitored the system signals and current levels. With this data displayed graphically, the success of the power control logic was observed immediately.

The initial state machine design and subsequent modifications went through the FPGA Design Flow as shown in Figure 2, *FPGA Design Flow.* The logic synthesis was controlled by user defined scripts and synthesis constraint files to produce an EDIF netlist. The Place and route tools imported the EDIF file and used minimal design constraints with fixed pin placements to generate the FPGA configuration bitstream. Each new bitstream was loaded into a new EPROM that replaced the one on the prototype board. Only the EPROM on the prototype board was replaced with each state machine modification. Retrieving and emulating an earlier design was as easy as replacing the EPROM on the prototype board with an alternate EPROM. In the future, download cable capability from the HP will further reduce the turn around time in configuring the FPGA.

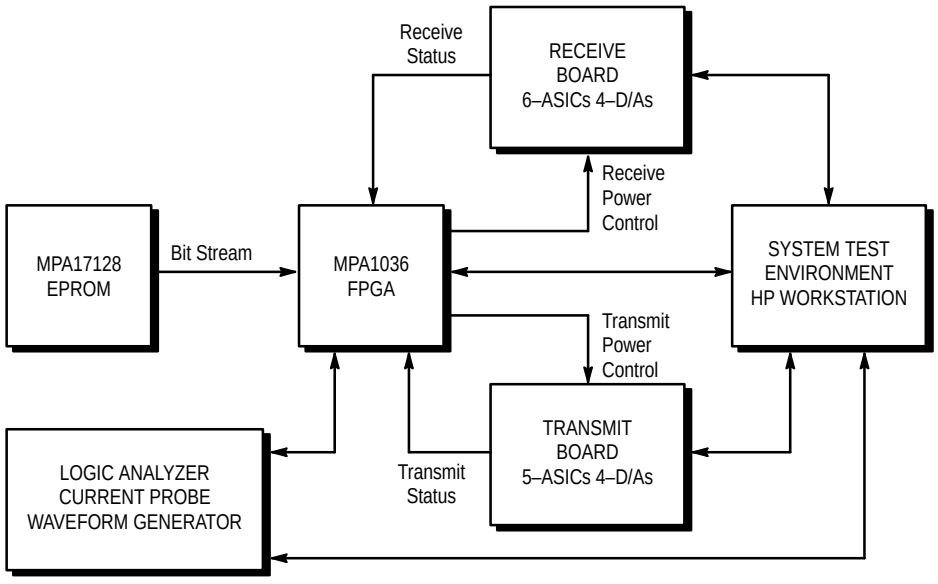

Figure 1. FPGA Prototyping Application

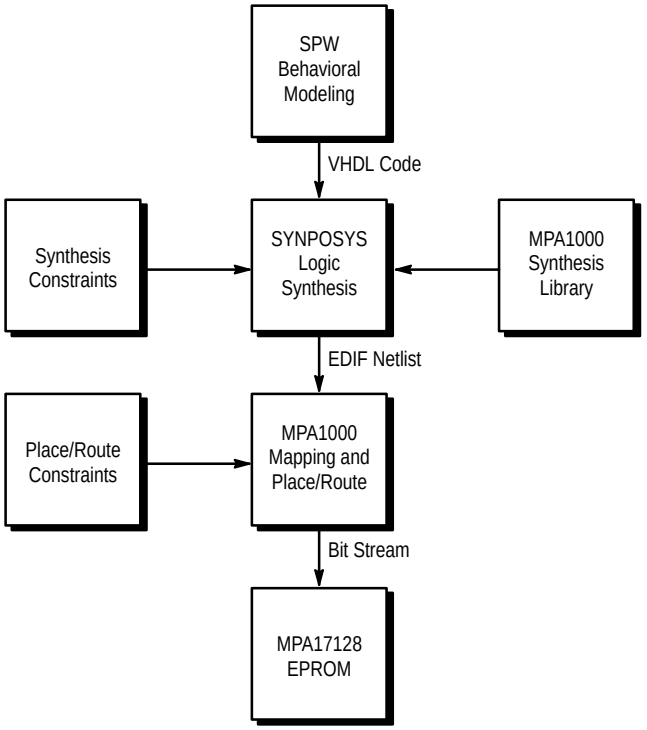

Figure 2. FPGA Design Flow

Summary

The development of the complex state machine that monitors and controls the power of the transmit and receiving modules of the communications system would be extremely time consuming and difficult to verify in a simulation environment. The rapid prototyping and emulation approach described offered a flexible design environment with quick turn–around time. In this case, new RTL code could be synthesized, placed and routed, and a new FPGA configuration bitstream generated in less than two hours. The advantages are the quick turn– around time for design modifications and the real time system emulation and verification. Down loading the FPGA configuration, performing the system emulation, and observing the system signals and current probe readings proved to be the quickest way to evaluate the control logic. The alternative would be to make several system level simulations that take typically two to three days per simulation, then several hours to evaluate the simulation data.

Choosing an SRAM based FPGA proved cost effective by being able to reprogram the existing FPGA. The EPROM that was used to hold the configuration bitstream costs approximately $7.00. An equivalent antifuse FPGA that would require a new part for every evaluation would cost approximately $70.00.

For the solution that met the design requirements, the HDL code was then merged into an existing ASIC HDL module without any technology mapping issues.

How to reach us:
USA/EUROPE: Motorola Literature Distribution;
P.O. Box 20912; Phoenix, Arizona 85036. 1–800–441–2447

MFAX: RMFAX0@email.sps.mot.com –TOUCHTONE (602) 244–6609
INTERNET: http://Design–NET.com

JAPAN: Nippon Motorola Ltd.; Tatsumi–SPD–JLDC, Toshikatsu Otsuki,
6F Seibu–Butsuryu–Center, 3–14–2 Tatsumi Koto–Ku, Tokyo 135, Japan. 03–3521–8315

HONG KONG: Motorola Semiconductors H.K. Ltd.; 8B Tai Ping Industrial Park,
51 Ting Kok Road, Tai Po, N.T., Hong Kong. 852–26629298