Testing and Debugging

EDAN85

Advanced Embedded Systems Design Course

Terminology

- Testing: making sure that we're building the system right.
  (Verification - building the right system)
- Debugging: if it does not, figure out the problem and solve it.
- Offline: ...while the system is not running (or running on a host)
- Runtime/online: ...on the running system (or running on target)

Methods overview

	Hardware	Software
Simple	signals-to-pins (LEDs, multimeter, oscilloscope)	code inspection, printouts
Complex, Tool based	simulator (ISim), on-chip signal capture (Chipscope)	host debugging, target debugging (xmd)

Simple, in hardware

Route simple (not bus) signals to free/certain FPGA pins:

- LEDs for signals that do not change often, or need to be mainly in one state or another, or just need to be checked that they are driven by someone: i.e. RESET signals
- use a voltmeter to check the level, or average value (PWM like behavior): CLOCK, PWM, GND, VCC,...
- use an oscilloscope/logic analyzer when exact timing is essential: VGA synchronization signals, CLOCK,...

Simple, in software

- use debug printouts to identify which phases the program passes through
  - is it entering **main**?
  - is the Hw setting up properly? (check error codes)
  - are the registers loaded with the right values? (write/read peripheral registers)
- use debug levels to separate messages

```
#define DBG(L, txt) if(DBGLVL >= L) print(txt)
```

- use the provided self-tests for the IPs, or write your own tests

Complex, in hardware

Offline method: VHDL/Verilog **simulation**
Strongly recommended!

- for custom hardware IPs (test+debug)
  (although possible for whole systems, it is not recommended)
- write your own VHDL test modules to cover as much as possible of the required behavior
- low level simulation (post place&route) can detect timing problems very difficult to discover otherwise!

Complex, in hardware

- Online signal monitoring: **Integrated Logic Analyzer**
  - easy to set up, in-system extra cores
  - simple signals or busses
  - JTAG based PC GUI
  - instant or (simple/repeated) trigger based sampling (with sequencers)
  - integration with software debugging/MDM

Essential tool for runtime debugging of custom hardware!

Complex, in software

1. in cross-platform development:
   debug using the environment on the host machine
2. debug on the target machine:
   - the processor must have support for debugging (exceptions)
   - peripherals must have support for debugging (freeze signal)
   - compile applications with debug information

Further reading/to do

- working with the MDM and ILA cores for debugging and monitoring signals
- debugging using SDK and Vivado Logic Analyzer

Follow the Xilinx Embedded Processor Design (UG940) Lab3 (…but adapt it to your system!)