Hardware Design for Cryptography

Francesco Regazzoni

Contents

Simplified Hardware Design Flow

- Start from the specifications (algorithm standard, requirements....)
- Express the specification in an Hardware Description Language (HDL)
- Synthesize the HDL using a target library (express your functionality using gates/memory elements belonging to the library)
- Place the selected gates on the available silicon area
- Connect the gates

Obtained results

- Depend on the HDL code....
- And on the tool version! (Also Significantly!)

Contents

Performance Parameters

- Area: total area occupied by the circuit
- Throughput: data processed per time unit
- Power: Peak power for operation
- Energy: Total energy to complete a task
- Time to Market: total time to complete the design
- Latency: time to process a given data item

How to Improve Performance

Use More Advanced Technology

- Easy way
- Not Always Feasible
- Cost Increases
- Other Problems (adapting, missing IPs)

Make a Better Implementation

- Hard (requires designers to invest time)
- Not Always Feasible (limit in what can be done)

EDA Tools are designed for Industry

Industry

- Circuit has to work in the worst case
- Constraints are used to define specification
- EDA tools ensure that the circuit performance meets specification

Academia

- Interested only on limits (smallest, fastest...)
- Performance numbers needed for comparison
- Not easy to get fair numbers from tools
- Constraints are “used” to explore design space

Academia Ignores some Problems

- Overhead is not always included (Large I/O bandwidth, Very fast clocks, Frequency scaling)
- Additional Verification Efforts (Interfaces between different clock domains,...)
- Testing overhead (difficult to test asynchronous designs)

Quality of Results Depends on Design Stage

- **Synthesis**
  - First level of Physical Properties
  - Routing overhead unknown
  - Timing and Power models are mean values
- **Post Layout**
  - Routing is know, more accurate timing
  - Not all post-layout include proper provision for testing and power
- **Actual Measurements**
  - Real proof of concept
  - Performance affected by practical problem, worst then expected
  - Require a costly test infrastructure

Area

How much silicon area will be used for the circuit?

- **Why Care**
  - Silicon Cost
  - Feasibility (will it fit?)
- **“Should” be easy to determine**
  - Does not change during/after fabrication
  - Reliable and accurate post-layout numbers
- **Units**
  - mm^2: correct, technology dependent
  - GE: commonly used, not accurate

Gate Equivalent

- Total Area divided by the 2-input NAND
- Coarse measure: varies **at least** 10% between different technologies

But what is the area?

- Numbers reported after synthesis (routing missing, power connections missing, clock missing,...)
- Post layout gate counts (still missing power, routing, ...)
- Smallest rectangle where the block fits?
- Total die area (area can be imposed mini@sic, if the chip has more than one design?)

Numbers are approximated

- Synthesis does not tell the whole story (routing overhead can be between 10% and 500% design dependent)
- Post layout numbers are more reliable
- Total chip is easy to determine (however, not often the whole chip is of interest)

Speed

How fast is my circuit?

- Latency: time required to perform some action, measured in units of time (nanoseconds, clock periods, ...).
- Throughput: the number of such actions executed per unit of time, measured in units of whatever is being produced.

- Critical path is path which creates longest delay
- Clock cycles = $\frac{1}{critical path}$

Power Estimation

- Specify Voltage and Frequency
- Specify how circuit activity is determined
- Specify how the power was measured
- Specify the operating conditions used for the estimation (corner)

Contents

Reconfigurable Devices

- Field Programmable Gate Arrays (FPGAs)
- Reconfigurable hardware devices
- Trade offs between ASIC and microprocessors
- Current progresses allow to store a complete SoC on FPGA

Advantages

- Non recurring engineering costs
- Reduced time to market
- Always the latest technology

High Level View

- Configurable blocks (look-up-tables)
- Configurable routing matrix
- Input/Output blocks
- Memory configuration
- Advanced processing elements (DSP, whole processors)

Contents

Cryptographic Hardware

- Well defined (we have golden model)
- Simple (based on few operations)
- Easy to test (few vectors guarantee a good coverage)

What Cryptographic Algorithm needs?

- Fast Speed (sometimes)
- Low Latency
- Low Area
- Low Power
- Low Energy

What an Hardware Designer wants?

- Not many exceptions
- Scalability (number of rounds, state size, word size multiple of 2)

Questions?

Thank you for your attention