CSE241 Project

THIS PROJECT IS IN TWO PARTS. BOTH PARTS MUST BE COMPLETED. DUE DATE IS FRIDAY OF 10TH WEEK.

Part A: Synthesis, Place-Route, and Optimization of Encryption Cores

Abstract

The goal of this project is to synthesize and optimize a given Verilog netlist into a functional, placed and routed design. The design is based on the Simple AES/Rijndael IP Core, found at http://www.esat.kuleuven.ac.be/~rijmen/rijndael/. This particular AES implementation is with a 128 bit key expansion module only.

You will need to write a Verilog top module that uses two AES cipher cores, then synthesize and place-and-route to given basic constraints. A golden set of numbers will be provided to permit benchmarking against the TA's own implementation.

Note: Design must be performed in groups of two. Design exploration maybe performed independently.

Basic Core Diagram

```
+--------------------------------+    +-------------------+    +-------------------+
|                                |    |                  |    |                  |
|                Control          |    |                Key|    |                Final|
|                done            |    |  Expansion      |    |  Permutation     |
|                                |    |  Round          |    |  Permutation     |
|                                |    |  Permutation    |    |                  |
|                                |    |  text_in        |    |  text_out        |
|                                |    |                 |    |                  |
```

Synthesizables

- aes_cipher_top.v
- aes_key_expand_128.v
- aes_rcon.v
- aes_sbox.v
- timescale.v

Signals

A single global clock signal is assumed, but the cores must be controllable as separate entities. E.g. the rst signal is not global, but local to each instance. See I/O section for full signal descriptions.
I/Os

<table>
<thead>
<tr>
<th>Cipher</th>
<th>Name</th>
<th>Width</th>
<th>Type</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>clk</td>
<td>1</td>
<td>I</td>
<td></td>
<td>core clock</td>
</tr>
<tr>
<td>rst</td>
<td>1</td>
<td>I</td>
<td></td>
<td>active low synchronous reset</td>
</tr>
<tr>
<td>ld</td>
<td>1</td>
<td>I</td>
<td></td>
<td>load</td>
</tr>
<tr>
<td>done</td>
<td>1</td>
<td>O</td>
<td></td>
<td>done</td>
</tr>
<tr>
<td>key</td>
<td>128</td>
<td>I</td>
<td></td>
<td>key</td>
</tr>
<tr>
<td>text_in</td>
<td>128</td>
<td>I</td>
<td></td>
<td>input text block</td>
</tr>
<tr>
<td>text_out</td>
<td>128</td>
<td>O</td>
<td></td>
<td>output text block</td>
</tr>
</tbody>
</table>

Libraries

The final implementation will be in Artisan TSMC .18um LV, [http://www.artisan.com](http://www.artisan.com). LSI 10K 1.0um libraries should be used to develop and debug tool scripts.

PART A Formal Specification

Your design must encipher 2 keys per each clocked input. The function will not be verified, but proper connectivity must be maintained at the top Verilog level. Care must be taken to avoid duplication of net names. No area target is specified, but gate count should be kept low (this is actually implicit in the objective function given below).

You will make 1 implementation of your design.

1. **PDA product driven**: Minimize the Power * Delay * Area product of the design (where Power is measured by a Perl script from DEF layout (script to be provided), Delay is 1/frequency, and Area is the core region area), subject to the constraint that Delay is at most 5ns.

A basic synthesis script is available for the AES core, [aes_synth_start.scr](http://www.artisan.com). Basic P&R scripts will also be provided for this part of the project, as well as for Part B. A .synopsys_dc.setup is also provided.

There are no pin-out requirements for this design. Pins may be placed randomly, or according to the constraints that benefit your design.

**Deliverables**

You must provide DRC and LVS clean versions of your designs in both (routed) DEF format and GDSII formats. Final timing (Synopsys PrimeTime), area and power reports are also required for your designs.

Part B: Placement, Routing and Optimization of AES cores

**Abstract**

The goal of this part of the project is to synthesize and optimize a given Verilog netlist into a functional, placed and routed design based on timing, power or area driven design. Your goal is to minimize one of the three targets.
PART B Formal Specification

Your design must encipher 2 keys per each clocked input, as in Part A. The design must meet given constraints and minimize power, delay, or area. The final layout must pass formal equivalence checking (netlist to netlist), as well as physical verification (LVS, DRC, final RCX). Final timing signoff is determined by Synopsys PrimeTime.

You will make 1 implementation of your design.

Remember that each design step must focus on the chosen objective; this includes synthesis and P&R constraints.

Your group must choose some of three aspects of the design and optimize it.
Choices available:
- Optimize area subject to a lower bound on clock frequency;
- Optimize power subject to a lower bound on clock frequency; or
- Optimize clock frequency subject to an upper bound on area.

You will need to use the combination of synthesis, placement and routing tools to achieve your objective. Each group must e-mail their objective to the TA for posting on the main web page. To achieve balanced competition, there can be at most 5 teams addressing any given objective.

Power

Power estimates will come from a supplied script. You will need to extract parasitics from the design with Assura RCX. This RSPF file along with a design activity profile (on/off periods for the design) will be used as input for this file.

Timing

Basic frequency minimums must be met for the design.

Area

Appendix

Recommended Tool Flow

Synthesis
Synopsys DC
- possible ultra optimization, but doubles resources and synthesis time
- intermediate storage as .db
- output: .sv
- optimize with worst case + 10% margin
- 2.5 ns period for .18u Artisan library (6 layer metal)
Primetime
- Initial timing check of core

Place and Route
SE DSM P&R
- Artisan tech LEF, Antenna LEF, TLF (timing driven)
- Qplace
- Ctgen
- Pbopt
- Power structures
- Wroute
- DEF
- SDF file gen
- GDSII

**Assura/Virtuoso/DIVA**
- DRC/LVS check
- RCX
- Formality
- Quick logic-equivalence check

**Primetime**
- Back annotate parasitics
- Re-check timing

**Optional**
Use back annotation for synthesis run, with no wire-load model for accurate simulation with parasitics.
- Re P&R

**Directory Structure**

**Main:**
ee260b/final_project
  / 
  part_a part_b

**AES**
aes_core
  / 
  bench rtl doc sim syn SE
  | 
  verilog bin scripts reports def gds log gcf
  / 
  synth log work out

**Libraries:**

**Artisan TSMC .18u**

**Synopsys DC Setup**
/home/solaris/ieng9/ee260b/public/final_project/aes_core/syn/bin/.synopsys_dc.setup

**Cadence Tool Start**
/home/solaris/ieng9/ee260b/public/final_project/aes_core/SE/.cad.cshrc