Advanced VLSI Design Project

ASIC Design of CAN Protocol Controller using OSU 180nm Technology

Student:
Alireza Haghdoost

Instructor:
Prof. Shahin Hessabi

Summer 2010
## Table of Content

- **CAN Protocol Description** 3
- **CAN Protocol Implementation** 3
- **CAN Protocol Testing** 3
- **Design Synthesis** 6
  - Synthesis for Area 7
  - Synthesis for Delay 8
- **Physical Design** 9
  - Floorplan for Area Efficient Circuit 10
  - Floorplan for Delay Efficient Circuit 11
- **Backannotation and Post-Layout Simulation** 12
  - Post-Layout Simulation for Area Efficient Circuit 13
  - Post-Layout Simulation for Delay Efficient Circuit 14
• CAN Protocol Description

Controller Area Network (CAN) is a bus protocol standard designed to for communication between microcontroller and other embedded device in an embedded system. CAN protocol need no host management node in a network and nodes can speak with each other independently.

Main applications of CAN protocol is in the modern automobile where about 70 electronic control units need in various subsystems need to communicate and sync with each other. For example a CAN bus may be used in cars to connect engine control unit and transmission or to connect door locks.

• CAN Protocol Implementation

In this project I used available CAN protocol IP core in the opencores\(^1\) community implemented with Verilog HDL, figure 1 shows architecture of implemented design.

![Figure 1: CAN protocol Implementation diagram](image)

• CAN Protocol Testing

I use a testbench application written in Verilog HDL to test implemented CAN modules in the next sections. The testbench have two instance of CAN protocol controller that are connected with each other with an InOut port. Testing done with simple send and receive issue command from instance 1 to instance 2. Figure 2 shows the block diagram of testbench module.

\(^1\) [www.opencores.org](http://www.opencores.org)
The packet that was written in the interface port between CAN instance 1 and 2 carefully monitored to satisfy correctness of CAN modules. I was interested to view the test packet shows in figure 3 for this work in the ModelSim Wave window. Figure 3 is the result of behavioral Simulation of CAN design.

![Figure 2: CAN Implementation testing](image)

**Design Synthesis**

For logic Synthesis I used Synopsys Design Compiler with OSU 180nm fabrication technology library. Synthesis of the CAN module done with two configurations for efficient area and delay. Following I will describe each configuration and post synthesis simulation results.

- **Synthesis for Area**
  
  I used set_max_area 0 in logic synthesis script to force Design Compiler optimize CAN design for best area.
Reported area utilization described in the following table:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of ports:</td>
<td>19</td>
</tr>
<tr>
<td>Number of nets:</td>
<td>9877</td>
</tr>
<tr>
<td>Number of cells:</td>
<td>9428</td>
</tr>
<tr>
<td>Number of references:</td>
<td>24</td>
</tr>
<tr>
<td>Combinational area:</td>
<td>229097</td>
</tr>
<tr>
<td>Noncombinational area:</td>
<td>167664</td>
</tr>
<tr>
<td>Total cell area:</td>
<td>396761</td>
</tr>
</tbody>
</table>

Reported timing described in the following table:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>data arrival time</td>
<td>5198.74</td>
</tr>
</tbody>
</table>

Post synthesis simulation also have done for the synthesis design for efficient area, as you can see in figure 4, synthesized netlist has correct output.

Figure 4: post Synthesis Simulation for area efficient design

- **Synthesis for Delay**

In logic synthesis for best delay, I have synthesis behavioral model with very high frequency to force Design Compiler optimize design for the best delay. Following is the reported area and delay:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of ports:</td>
<td>19</td>
</tr>
<tr>
<td>Number of nets:</td>
<td>10310</td>
</tr>
<tr>
<td>Number of cells:</td>
<td>9864</td>
</tr>
<tr>
<td>Number of references:</td>
<td>26</td>
</tr>
<tr>
<td>Combinational area:</td>
<td>237895</td>
</tr>
<tr>
<td>Noncombinational area:</td>
<td>167664</td>
</tr>
<tr>
<td>Total cell area:</td>
<td>405559</td>
</tr>
</tbody>
</table>

Reported timing described in the following table:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>data arrival time</td>
<td>218.9</td>
</tr>
</tbody>
</table>

Post synthesis simulation also has done for the synthesis design for efficient area, as you can see in figure 5, synthesized netlist has correct output.
Physical Design

After logic synthesis with Synopsys Design Compiler, I have used Cadence SoC encounter for physical design and produce final layout in GDSII format.

- **Floorplan for Area Efficient Circuit**

  Cadence SoC encounter, propose 770x770 die. For physical design of area efficient circuit, however, for better placement and routing first I use 800x800 silicon die. After placement and nanoroute I found that my design has negative slack for setup time. The negative slack in 800x800 die will not solve with optimization neither for 900x900 die size, so I chose 1000x1000 die size for best area efficient circuit. Figure 6 shows the floorplan and figure 7 shows the layout of the implemented design:
Figure 5: Floorplan for area efficient design

Figure 6: Physical layout for area efficient design

- **Floorplan for Delay Efficient Circuit**
  Physical design for the delay efficient design also done in 1000x1000 die size and all timing constrains passed in the first timing optimization try. Figure 7 and 8 shows floorplan and physical layout of implemented circuit for delay efficient design.
• Backannotation and Post-Layout Simulation
  Generated Verilog netlist from Cadence SoC encounter is capable to backannotate to SPICE netlist using HSIM v2s function. I use combination of HSIM and ModelSim for post layout simulation. In the following I describe the simulation results for each circuit:
  o Post-Layout Simulation for Area Efficient Circuit
As shown in figure 9 the simulation wave forms in combination of HSIM and Modelsim is similar to past simulation waves in post-synthesis and pre-synthesis simulations.

Figure 9: post layout simulation for area efficient circuit

- **Post-Layout Simulation for Delay Efficient Circuit**

As shown in figure 10 the simulation wave forms in combination of HSIM and Modelsim is similar to past simulation waves in post-synthesis and pre-synthesis simulations.

Figure 10: post layout simulation for delay efficient circuit