Design Challenges: An Industrial View

- Correct functionality requirements

Design Challenges:
- Correct functionality requirements
- Timing
- Area
- Power
- Runtime
- Memory
- Time to Market

Designer

EE M216A :: Fall 2011

Hardware Description Language (HDL) and Logic Synthesis

Alireza Tarighat (ee216a@gmail.com)

- Slides are Courtesy of Prof. Dejan Marković.

UCLA
Logic Synthesis

- **Synthesis Tool**
  - Optimizes gate-level implementation for speed/area/power
    - Sizing the gates, inserting gates, etc.
  - Does not touch functional behavior (maintains logic equivalency)
  - Works with any standard cell library (any technology, any foundry)
  - Decouples logic design from technology/fab/node and standard cell library
  - Takes in different input formats
    - VHDL
    - Verilog
    - Gate-level netlist
  - Takes in different design constraints
    - Simplest one: clock period

- **Design constraints**
  - Environmental constraints
    - Driver
    - Load (max fanout)
  - Define clocks
    - Cycle time
    - Uncertainty (jitter)
  - Optimization constraints
    - In and out delay / max transition
    - Max area etc.
    - Clock period vs. area vs. power
  - Timing exceptions
    - Multi-cycle, False-path
    - Example Synthesis Tool:
Logic Synthesis is Timing-Driven

- This is a generic design used during synthesis
  - Internal data-path delay (cycle (and hold) time analysis)
  - Relationship to in and out paths
  - Timing exceptions

![Diagram]

Timing Constraints

- **Clock period**: set by `define_clock`
- **Input delay**: arrival of an external path with respect to a Clk edge
- **Output delay**: timing path from an output port to a register input of an external block

- Input and output delays budget timing for surrounding logic in general case when the in/out ports are not registered
Understanding Timing Constraints

- Three important constraints (clock, input, output)
  - Blue box = current_design (to be retimed)

```
logic
```

```
set_input_delay
```
(affects input logic)

```
set_output_delay
```
(affects output logic)

```
create_clock
```
(affects internal logic)

Environment: Drivers, Load...

- To simulate realistic inputs, we can set the driving cell using the `external_driver` command, to be any cell in the library
  - This ensures that the input has a finite slew rate

- The load capacitance can be set on the output ports using the `external_pin_cap` command

- By default the Encounter RTL Compiler only tries to meet the timing constraints without optimizing power

- If the `max_dynamic_power` attribute is set to some value, the tool tries to meet the timing specs while also optimizing for power in the process
Example: Automated Adder Synthesis

- Copy the following files to your work directory
  /usr/public.2/ee216a/cadence/SOC62/SynLib.v
  /usr/public.2/ee216a/cadence/SOC62/adder.v
  /usr/public.2/ee216a/cadence/SOC62/adder.tcl

- The top level synthesis script adder.tcl reads in the HDL file, sets timing, load and power constraints, and runs synthesis

- To run RTL synthesis type the following command
  > rc –files adder.tcl –gui

- The GUI window will show detailed architecture (gate level). Use report power, report area, report timing commands in the rc command window to get Power, Area and Delay numbers

---

Setup Example: Adder.tcl (1/2)

```
set_attribute library /w/apps/apps.16/cadence/gsclib090_v2.9/timing/typical.lib      Define lib
set_attribute hdl_language v2001
read_hdl adder.v
read_hdl SynLib.v
elaborate adder

define_clock -name clk -period 1000 -design /designs/adder
  {/designs/adder/ports_in/clk}
dc::set_time_unit -picoseconds

dc::set_load_unit -femtofarads

dc::set_input_delay 20 -clock clk [all_inputs]
dc::set_output_delay 100 -clock clk [all_outputs]

set_attribute external_driver [find [find /libcell DFFX1] -libpin D]
  {/designs/adder/ports_in/*}
set_attribute external_pin_cap 26.5488 {/designs/adder/ports_out/*}

set_attribute lp_power_unit mW /
set_attribute max_dynamic_power 0.5 /designs/adder
```

Set timing
Set env.
Opt.
Setup Example: Adder.tcl (2/2)

How does synthesis work?

Example Clock, In and Out Delay [Synopsys]

Define clock, input and output delay

```
cREATE_CLOCK -NAME "Clk" -PERIOD $Tclk
SET_INPUT_DELAY 0.5
SET_OUTPUT_DELAY [EXPR $Tclk - 0.5 + $wire_margin] ;# wire_margin = 0.2
```

Work with expressions and variables
The adder.tcl file sets constraints & optimization parameters

```tcl
set_attribute library /u/apps/apps.16/cadence/gcлюбv2.9/timing/typical.lib
set_attribute hdl_language v2001
read_hdl adder.v
read_hdl SynLib.v
elaborate adder
dc::current_design adder
dc::set_time_unit -picoseconds
dc::set_load_unit -fFIM"
define_clock -name clk -period 500 -design /designs/adder (/designs/adder/ports_in/clk)
dc::set_input_delay 20 -clock clk [all_inputs]
dc::set_output_delay 100 -clock clk [all_outputs]
set_attribute external_driver [find [find / -libcell OFFxy] -liboin D] [/designs/adder/ports_in/x]
set_attribute external_pin_cap 26.5488 [/designs/adder/ports_out/x]
set_attribute ip_power_unit mw
set_attribute max_dynamic_power 0.25 [/designs/adder]
synthesize -toMapped -effort high
report_area > adder_area.rpt
report_power > adder_power.rpt
report_timing > adder_timing.rpt
report_clocks > adder_clocks.rpt
```

---

Pre- and Post-Layout Clock

![Pre- and Post-Layout Clock Diagram](image)

- **Pre-layout**
  - `create_clock -p 30 -n MCLK Clk`
  - `set_clock_uncertainty 0.5 MCLK`
  - `set_clock_transition 0.25 MCLK`
  - `set_clock_latency -source 4 MCLK`
  - `set_clock_latency 2 MCLK`

- **Post-layout**
  - `create_clock -p 30 -n MCLK Clk`
  - `set_clock_uncertainty 0.1 MCLK`
  - `set_clock_latency -source 4 MCLK`
  - `set_propagated_clock MCLK`
Synthesis Results: Fast Adder

- Adder synthesized to meet timing constraint of 550 ps
- Input is registered from a FF
  - Clk-Q delay = 157 ps
  - Setup time = 82 ps
- Effective adder delay = 311 ps
- Timing constraints are stringent, tool synthesizes a carry look ahead type of adder
- Synthesis reports
  - Area = 1035 μm²
  - Energy (active) = 0.2145 fJ
  - Power (leak) = 0.005 mW

Synthesis Results: Slow Adder

- Adder delay = 757 ps
- Structure is somewhat like carry ripple topology
- Area = 548 μm² (half of previous)
- Energy (active) = 0.13 fJ
  - Nearly ½ of previous design
  - Expected since \( V_{DD} \) was the same, \( E_{active} \) depends only on \( C_{sw} \), which was halved with the area being halved...
- Power (leak) = 0.002 mW
  - Also reduced due to reduced area
Synthesis Results

- Energy-delay tradeoff plot obtained from synthesis
  - Keep timing constraint, move down the energy axis (left plot)
  - Resulting energy-area should be below reference curve (right plot)

High-Level Design Issues

- You may think design is a straightforward logical process
  - Start with the idea of what you need to build
  - And then build it
- Real design is not like that
  - Think you have an idea of what to build
  - Through the design process you figure out what you really want to build
  - Need to validate idea early in the process
- What you build depends on the implementation capabilities and constraints
  - Implementation issues will change the specification

Need a language that helps with the real (interactive) design process