Project Timing Analysis

Jacob Schneider, Intel Corp
Sanjeev Gokhale, Intel Corp
Mark McDermott
Overview

• Brief overview of global timing

• Example of extracting AT, RAT, and PASSTHROUGHs from a block of verilog

• What does the timing team do?

• What do you need to provide to the timing team?
Basics of Timing

**Required Arrival Time (RAT)**

**Internal Flop 2 Flop**

**Arrival Time (AT)**

Clock capture time = Clock Period – Clock uncertainty = 900ps currently
Basics of Timing (cont)

Delay through Passthrough Block = input path delay + comb delay + output wire delay
Calculating Wire Delay

- Wire Delay can be calculated using provided Wire Calculator (need someone to work on this 65nm)

- Block Owners can use only M1, M2, M3 and M4 to route signals in their blocks. NOTE: M1 and M2 can be routed horizontally and vertically.

- Metal 5 and Metal 6 is reserved for Inter Block Routing and will be available for data path blocks. M7 and M8 are used for clocks and power routing.

- Use wire width and spacing of 1 unit for most instances.
  - Assuming there is space, wires of wider widths or spaces will be allowed, but first cut timing will assume minimum width/spacing.

- First pass required arrival time and arrival times need not include wire delay, as pins will be assumed to be in the middle of blocks. As pins move out to block boundaries, wire delay will need to be included in the RATs and ATs.
Calculating Combinational Delay

• All characterization data Combinatorial Gates is provided for the class.

• Please refer to Library Spreadsheets in VLSI 2 Project Page

• For all combinational logic driving output pins, assume a fanout of 3 (for delay purposes).

• You may change the size of your gates to be able to use a smaller fanout number, but any gates that are upsized must be accounted for in your spreadsheet (in number of minimum transistors).
Calculating Setup Times and C2Q Delay

- All characterization data for Flip Flops is provided for the class.

- Setup Time = TBD ps
- Clk2Q Time = TBD ps
  - These are values that may change as we investigate other types of flops and verify timing.

- Assume the input capacitance for a FF is 2x that of an inverter for fanout purposes.
What the Global Timing Team needs?

- Global Timing Team needs RAT/AT and Pass through Path Delays
  - RAT and AT times are relative to a clock cycle, while passthroughs are absolute delays
- Provide Global Timing Team with an <Block>.atrat file of following format
  - BLOCK_NAME  <Block Name>
  - START_RAT_SECTION
    - PIN  <Input Sig1>  clk  rise  <RAT1>
    - PIN  <Input Sig2>  clk  rise  <RAT2>
  - START_AT_SECTION
    - PIN  <Output Sig3>  clk  rise  <AT3>
    - PIN  <Output Sig4>  clk  rise  <AT4>
  - START_PASS THROUGH_SECTION
    - PASS THROUGH <Input Sig5>  <Output Signal6>  <PT56>

- Global Timing Team is not responsible for Internal (Flop 2 Flop) timing.
- It is up to the individual block owners to meet timing within their blocks.
Pin File for Each Block

• For each block, we need a file called inst_<block>.pin that contains the pin location relative to the origin (bottom left corner) of your block.

For now, the location should be ½ height and ½ width.

The file will look like this...

FILENAME: inst_b1.pin
INSTANCE inst_b1
PIN <signal0> <x-coord> <y-coord>
PIN <signal1> <x-coord> <y-coord>
....

If your block dimensions were 200x400, your x-coord would be 100 and y-coord would be 200.
Floorplan File from Integration

• We need a floorplan file with all instance names and the location of the lower left corner of each instance, relative to the lower left corner of the chip.

• It should look like below…

```
INSTANCE_LOCATION       io_block  <xcoord>  <ycoord>
INSTANCE_LOCATION       iwb_biu    <xcoord>  <ycoord>
INSTANCE_LOCATION       iwb_biu    <xcoord>  <ycoord>
INSTANCE_LOCATION       dwb_biu    <xcoord>  <ycoord>
```
Example

- We will walk through the below code to show how to calculate passthroughs, RATs and ATs.

- input du_stall;
- input icpu_ack_i;
- input icpu_err_i;
- input flushpipe;
- output genpc_freeze;

- reg flushpipe_r;

- assign genpc_freeze = du_stall | flushpipe_r;

- always @ (posedge clk or posedge rst)
  - if (rst)
    - flushpipe_r <= 1’b0;
  - else if (icpu_ack_i | icpu_err_i)
    - flushpipe_r <= flushpipe;
  - else if (!flushpipe)
    - flushpipe_r <= 1’b0;
Example – Arrival Time

• Computing Arrival Times

flushpipe_r is launched by a flop. Clock2Q delay is 134.7ps
flushpipe_r goes through the same structure we timed before! (72.28ps)
Total Arrival Time is C2Q + Logic Delay + Wire Delay = 134.7 + 72.28 + WD
AT for genpc_freeze is 206.98ps (ignoring wire delay)
Example – Required Arrival Time

• Computing required arrival times

RAT for icpu_err_i and icpu_ack_i includes delay through a NOR, Inv, Mux, as well as the setup time to a flop.

RAT for flushpipe includes 2 mux delays and the setup time (use the worst case here, since flushpipe has 2 paths to the flop).

Since this path is receiving, assume the gates are minimum sizes.

Since we won’t have the nice fanout of 3 working for us in this case, it’s time for some Logical Effort fun! (Also applies to Arrival Time calculations.)
Example – Required Arrival Time

• Computing required arrival times

<table>
<thead>
<tr>
<th>Cell</th>
<th>Cin</th>
<th>G</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOR2</td>
<td>5</td>
<td>9</td>
<td>37</td>
</tr>
<tr>
<td>INV</td>
<td>3</td>
<td>9</td>
<td>21</td>
</tr>
<tr>
<td>MUX2</td>
<td>6</td>
<td>8</td>
<td>32</td>
</tr>
</tbody>
</table>

• The g and p values for the NOR, Mux, and Inv are listed in the table to the right.
• To use, multiply ‘g’ by ‘h’, which is the Cout/Cin value, and add ‘p’
• The NOR has h=3/5 (inv/nor), so its formula is 9*3/5+37 = 42.4ps (note that it’s larger than the FO3 value in the spreadsheet -> this shows that logical effort is not quite accurate…)
• The inv is driving a minimum mux, so its h is 6/3 (mux/inv). Delay = 9*2+21 = 39ps
• The mux is driving a flop (assume cin=6), so its h is 6/6. Delay = 8*1+32 = 40ps
Example – Required Arrival Time

• Computing required arrival times

- For icpu_err_i, the RAT is Clock Period - NOR - INV - MUX - Setup = 900 - 42.4 - 39 - 40 - 100 = 678.6ps
- icpu_ack_i sees the same path, so it’s RAT is also 678.6 ps
- flushpipe sees Clock Period - MUX - MUX - Setup = 900 - 40 - 40 - 100 = 720ps

*NOTE that again, these do not include any wire delay!!!*
Example – Internal F2F Path

You need to verify that all internal paths meet timing as well.

In this case you would make sure that the C2Q + Mux Delay + Mux Delay + Setup time is less than the clock period (900ps)

\[ \text{Delay} = 134.7 + 40 + 40 + 100 = 314.7 \text{ ps} < 900 \text{ ps} \rightarrow \text{In this case we meet timing} \]
Update the ATRAT file

- Now that we’ve got all of our times, we need to update our ATRAT file
- It should now look like the following

  - BLOCK_NAME  or1200_freeze
  - START_RAT_SECTION
    - PIN  icpu_ack_i  clk  rise  678.6
    - PIN  icpu_err_i  clk  rise  678.6
    - PIN  flushpipe  clk  rise  720
  - START_AT_SECTION
    - PIN  genpc_freeze  clk  rise  206.98
  - START_PASS_THROUGH_SECTION
    - PASS_THROUGH  du_stall  genpc_freeze  72.28
General Guidelines

• Need to have a RAT or PASSTHROUGH for all input pins (you may have both and you may have multiple RATS or PASSTHROUGHs)

• Need to have an AT or PASSTHROUGH for all output pins (again, you may have both and also multiple ATs or PASSTHROUGHs)

• All assign statements are combinational in nature, so they are PASSTHROUGHs unless all the variables are state variables

• All always @ (posedge clk … ) blocks are sequential and provide conditions for setting a state variable. If they receive an input pin there will be a RAT. If they drive an output pin, there will be an AT

• Any always @ block that does not have clk involved is again combinational. It would synthesize into a latch, but this is wrong. It is combinational and will have a PASSTRHROUGH unless all elements are state variables
What the Global Timing Team will provide?

- Generate top level connectivity and pin location file from floorplan file, pin files, and pin connectivity files

- Use the RAT/AT/Pass Through Timing Information provided by cluster to calculate slack for paths between blocks

- Slack = RAT - AT – Inter Block Wire Delay

- If Slack is negative, timing is not met and we will iterate through this again.

- It is up to Block Owners to work with each other and Timing and Integration Groups to fix any paths with negative slack.

- Obviously, the goal is to focus on paths with negative slack
First Pass Global Timing

• For the first pass, Timing Group will use FloorPlan provided by Integration Team

• Timing Group will use empty ATRAT files (no timing data from cluster)

• Timing Group will assume that all pins for a block will be exactly in center of the block.
  – i.e. Pin Position = Block Coordinates + \( \frac{1}{2} \) (Ver and Hor Dimensions of Block)
  – This means that you don’t need to include wire delay in your RAT, AT, and passthrough delays, although you will need to at a future date.

• A mock timing report will be created to make sure all tools are working

• Target is to deliver one week after initial floorplan file is available.
Future Timing Reports

• Subsequent timing reports will need timing information from blocks

• Pins will eventually be migrated to the borders of blocks
  – You will then need to update your pin files
  – You will also need to include wire delay in your timing files

• Suggestion: Create a timing spreadsheet that has columns for clock capture time, flop setup time, flop clock to q delay, logic delay, and wire delay that automatically calculates all of your timing paths for all of your pins
  – These values will most likely change a few times during the project, and this will allow you to quickly update your timing files