EE671: VLSI Design

Assignment 4

Sai Saketika Chekuri
190070054

October 20, 2022

The following delays are used for the logic gates required.

Function	Delay (in ps)
AND	44
A + B.C	58
XOR	68
A.B + C.(A + B)	68

Design

We need the G and P signals for all levels, which are calculated as below:

Stage 0

We need 16 G^0 and P^0 values, generated from each bit of input operands.

- $G^0(0) = A(0).B(0) + C_{in}.(A(0) + B(0))$
- $G^0(i) = A(i).B(i)$ for $i \neq 0$ and $i = 1, 2 \cdots 15$
- $P^0(i) = A(i) \oplus B(i)$ for $i = 0, 1, 2 \cdots 15$

Stage 1

We need 8 G^1 and P^1 values, generated from G^0 and P^0 values

- $G^1(i) = G^0(2i + 1) + P^0(2i + 1).G^0(2i)$ for $i = 0, 1, 2 \cdots 7$
- $P^1(i) = P^0(2i + 1).P^0(2i)$ for $i = 0, 1, 2 \cdots 7$

Stage 2

We need 4 G^2 and P^2 values, generated from G^1 and P^1 values

- $G^2(i) = G^1(2i + 1) + P^1(2i + 1).G^1(2i)$ for $i = 0, 1, 2, 3$
- $P^2(i) = P^1(2i + 1).P^1(2i)$ for $i = 0, 1, 2, 3$

Stage 3

We need 2 G^3 and P^3 values, generated from G^2 and P^2 values

- $G^3(i) = G^2(2i+1) + P^2(2i+1).G^2(2i)$ for $i = 0$
- $P^3(i) = P^2(2i+1).P^2(2i)$ for $i = 0, 1$

Stage 4

We need G^4 value since it represents the final carry, generated from G^3 and P^3 values

- $G^4 = G^3(1) + P^3(1).G^3(0)$

Carry Generation

All carries indexed by powers of two are equivalent to $G(0)$s of the stages. c_{int} is a 17-bit array representing all the internal carries.

- $c_{int}(0) = C_{in}$
- $c_{int}(1) = G^0(0)$
- $c_{int}(2) = G^1(0)$
- $c_{int}(4) = G^2(0)$
- $c_{int}(8) = G^3(0)$
- $c_{int}(16) = G^4$

The odd carries are generated from the above by using G^0 and P^0 values.

- $c_{int}(3) = G^0(2) + P^0(2).c_{int}(2)$
- $c_{int}(5) = G^0(4) + P^0(4).c_{int}(4)$
- $c_{int}(7) = G^0(6) + P^0(6).c_{int}(6)$
- $c_{int}(9) = G^0(8) + P^0(8).c_{int}(8)$
- $c_{int}(11) = G^0(10) + P^0(10).c_{int}(10)$
- $c_{int}(13) = G^0(12) + P^0(12).c_{int}(12)$
- $c_{int}(15) = G^0(14) + P^0(14).c_{int}(14)$

The remaining are calculated as follows:

- $c_{int}(6) = G^1(2) + P^1(2).c_{int}(4)$
- $c_{int}(10) = G^1(4) + P^1(4).c_{int}(8)$
- $c_{int}(14) = G^1(6) + P^1(6).c_{int}(12)$
- $c_{int}(12) = G^2(2) + P^2(2).c_{int}(8)$

The final $C_{out} = c_{int}(16)$

Sum Generation

Each sum bit is calculated as follows, for $i = 0, 1, 2 \cdots 15$:

$$sum(i) = P^0(i) + c_{int}(i)$$

Testbench

The trace file has two vectors, one with appended inputs and the other with carry-out and sum appended.

- Vector 1: B A Cin (33 bits)
- Vector 2: Cout Sum (17 bits)

This appending is done by DUT, which instantiates a Brent_Kung_adder component.

The testbench py file generates 'TRACEFILE.txt', which contains 10 randomized test-cases.

Results

- Delay in RTL simulation $\approx 345ps$
- Delay in Gate Level simulation $\approx 17ns$

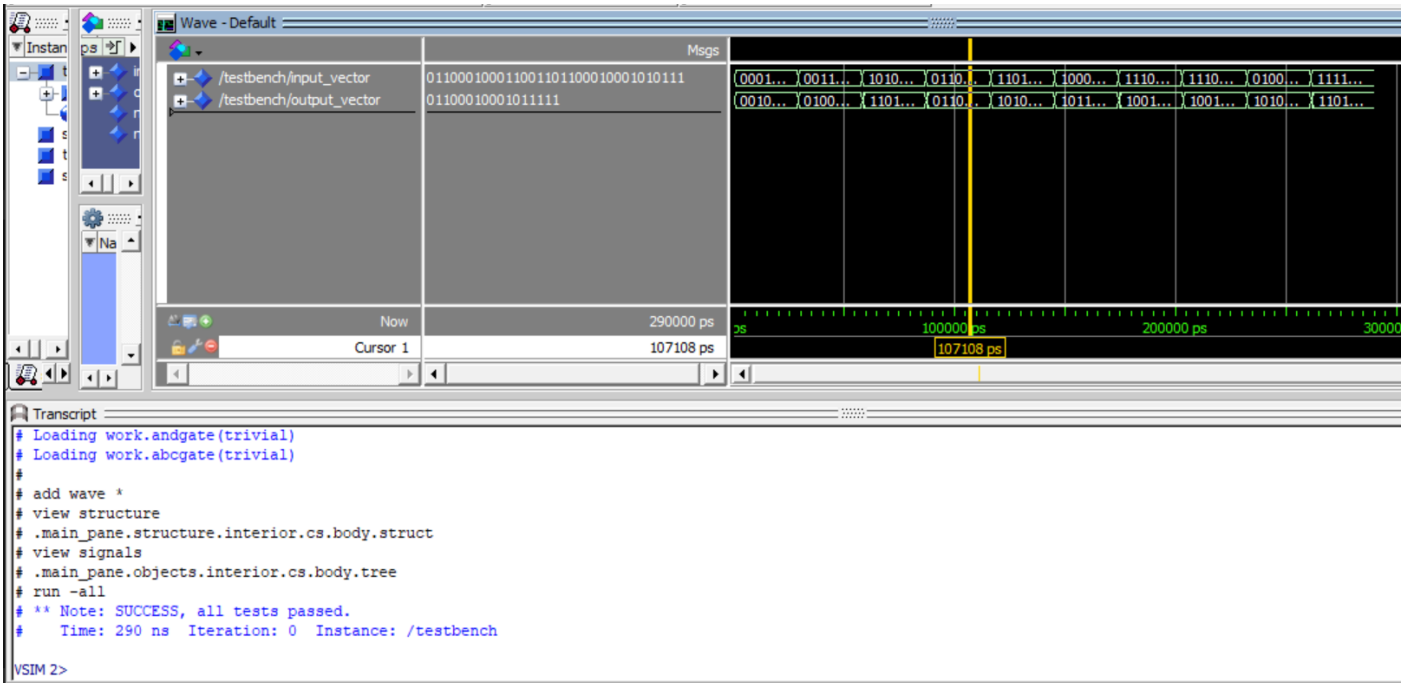

Figure 1: RTL Simulation

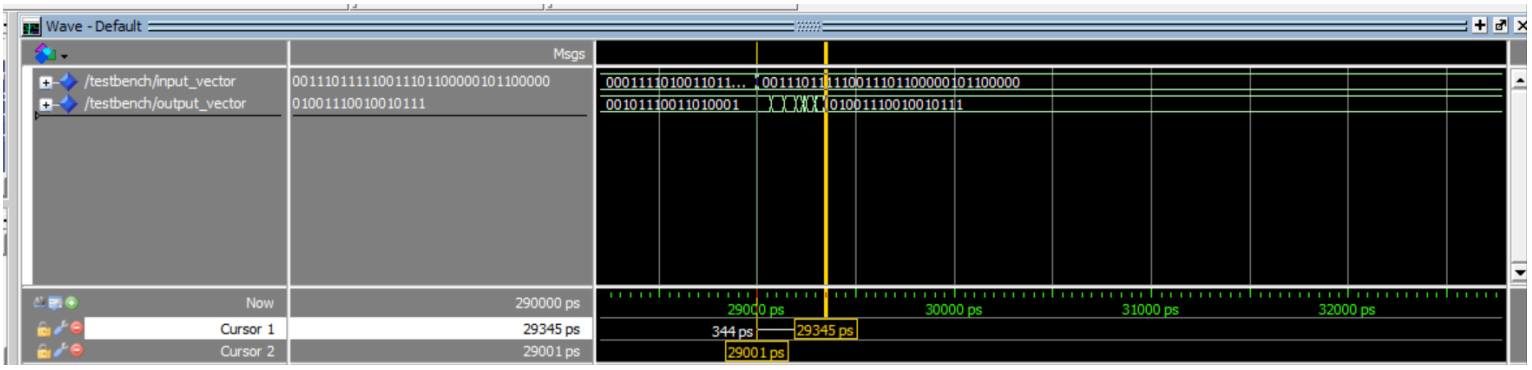

Figure 2: Delay in RTL Simulation

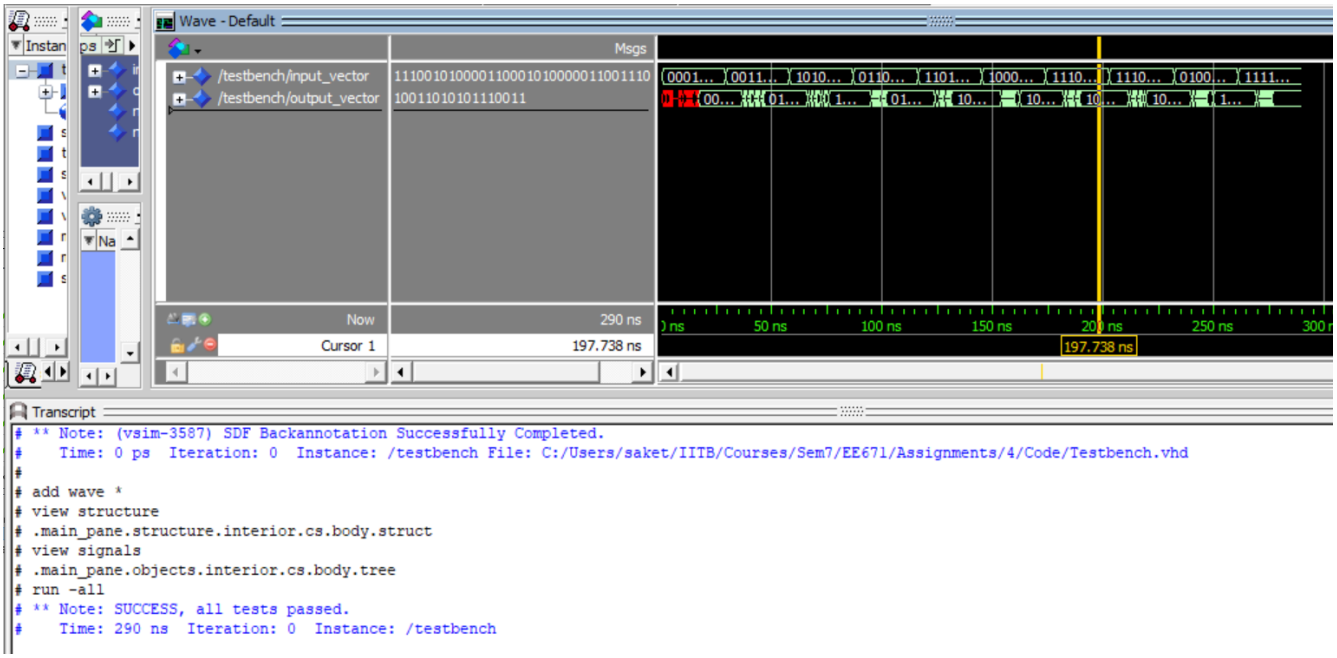

Figure 3: Gate Level Simulation

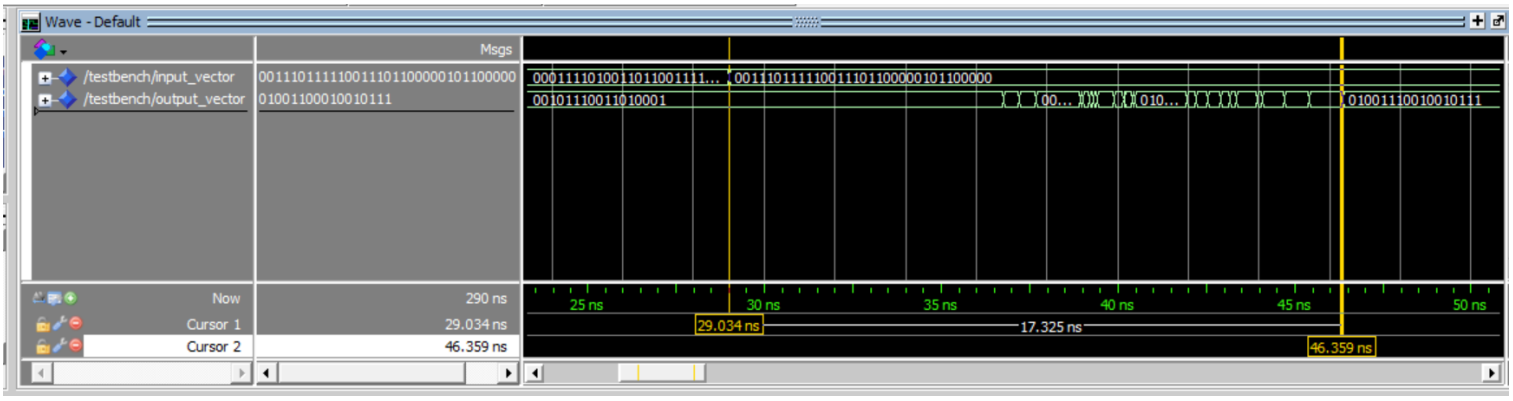

Figure 4: Delay in Gate Level Simulation