

# Weekly Report for Yu Hu's work in week3

January 30, 2005

## 1 Work1: Schedule my dual-Vdd buffer insertion work

The first work of this week is to schedule my dual-Vdd buffer insertion work. The goal of my project is to find an efficient approach to construct low-power dual-Vdd buffered tree, which can achieve optimal solution in a shorter running time (than dac05\_dualVdd paper). As a planning, my work can be accomplished with the following steps,

- Step 1. Prove/disprove the optimal for the constraint, which allows that only high-Vdd buffers drive low-Vdd ones. I calculate two typical cases to make a proof/disproof, which will be introduced later.
- Step 2. Read Shi's buffer insertion works (3 papers: DAC03, ASPDAC04, ASPDAC05) carefully, and extend these work to handle triple (power, RAT, capacitance) candidates cases.
  - (a) Firstly consider the problems which give the tree topology,
  - (b) Then consider simultaneously buffer insertion and tree construction.

## 2 Work2: Model and calculate delay-Power comparison of dual-Vdd cases

In this work, the main conclusion is that, testing with min-size devices parameters, it can keep optimum by only allowing high-Vdd buffers drive low-Vdd ones, when largest wire length of unbuffered interconnect is less than  $100\mu m$ . Otherwise, the optimum could be unacceptable. Details are described as follows.

As mentioned in step1 in section1. I'm going to calculate and compare the delays and power dissipations of the following two cases,

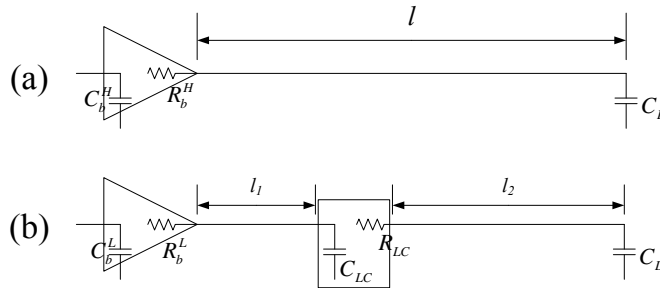


Figure 1: The two cases considered for the comparison

As shown in Fig.1, suppose that there exist some high-Vdd buffers in the downstream sub-tree, which is denoted by a load capacitance  $C_L$  in both cases. Case(a) shows the situation that allows only high-Vdd buffers drive low-Vdd ones, while case(b) shows the situation that a low-Vdd buffer and a level converter drive the sub-tree. In our comparison, I consider an ideal case, in which  $l = l_1 + l_2$ , where  $l$ ,  $l_1$  and  $l_2$  are wire length from sub-tree root to high-Vdd buffer, the one from level converter to low-Vdd buffer, and the one from from sub-tree root to level converter, respectively. To make the presentation clear, let  $l_1 = \alpha l$  and  $l_2 = (1 - \alpha)l$ , where  $\alpha \in [0, 1]$ .

Let  $Q_1$  and  $Q_2$  be delays in case(a) and case(b), respectively. Using the Elmore delay and  $\pi$  model in our calculation, we have

$$\Delta Q(\alpha) = Q_2 - Q_1 = A\alpha^2 + B\alpha + C \quad (1)$$

where  $A$ ,  $B$  and  $C$  are as follows,

$$A = rcl^2 \quad (2)$$

$$B = (R_b^L - R_{LC})cl + (C_{LC} + C_L - cl)rl \quad (3)$$

$$C = (C_b^L - C_b^H)R_0 + (d_b^L + d_{LC} - d_b^H) + R_b^L C_{LC} + (R_{LC} - R_b^H)C_L + (R_{LC} - R_b^H)cl \quad (4)$$

Let  $P_1$  and  $P_2$  be power dissipation in case(a) and case(b), respectively. Using the power modelling used in King Ho's dac05\_dvdd paper, we have

$$\Delta P(\alpha) = P_2 - P_1 = (P_{LC} + P_B^L - P_B^H) + 0.5\alpha(V_L^2 - V_H^2)cl \quad (5)$$

Obviously, the optimum of allowing only high-Vdd buffers drive low-Vdd ones can be proved if we can hold both  $\Delta Q(\alpha) > 0$  and  $\Delta P(\alpha) > 0$  for any  $\alpha \in [0, 1]$  under all typical settings for buffers and level converters.

I'm going to use the typical settings in 130nm, 90nm and 65nm technology nodes, respectively. In each technology node, the following (see Tab.1) parameters are needed.

Table 1: Parameters denotation list

$r$	Resistance of unit wire length of global interconnect with min width and space
$c$	Capacitance of unit wire length of global interconnect with min width and space
$C_b^H$	Input capacitance of high-Vdd buffer
$R_b^H$	Output resistance of high-Vdd buffer
$d_b^H$	Intrinsic delay of high-Vdd buffer
$P_b^H$	Dynamic power dissipation of high-Vdd buffer
$C_b^L$	Input capacitance of low-Vdd buffer
$R_b^L$	Output resistance of low-Vdd buffer
$d_b^L$	Intrinsic delay of low-Vdd buffer
$P_b^L$	Dynamic power dissipation of low-Vdd buffer
$C_{LC}$	Input capacitance of level converter
$R_{LC}$	Output resistance of level converter
$d_{LC}$	Intrinsic delay of level converter
$P_{LC}$	Dynamic power dissipation of level converter
$R_0$	Input resistance of driver

I tried to get some typical settings from ITRS03, but failed to find what I need. At present, I only get a group of settings under 65nm in table1 in King Ho's dac05\_dvdd paper, besides, for the level converter, the effective resistance, capacitance and intrinsic delay are 4733 $\Omega$ , 0.46fF and 220.1ps, respectively. I use the minimal size devices (buffers and level converter) in my calculation, and I set the largest length of unbuffered interconnect is 500 $\mu m$ , and  $C_L < c \times 500\mu m + C_{buffer}$ .

Based on the above settings, the coefficients  $A, B, C$  are approximated as  $A \approx 0.01l^2$ ,  $B \approx -2178l$ , and  $C \approx 245800 + 42598C_L$ . It's easy to see that  $\Delta Q(\alpha) > 0$  when  $\alpha < -B/(2A)$ . So if  $\Delta Q(1) > 0$ , then we can make sure that  $\Delta Q(\alpha) > 0$  when  $\alpha \in [0, 1]$ . In fact,  $\Delta Q(1) \approx A + B + C = 0.01l^2 - 2178l + 245800 + 42598C_L$ .

**CONCLUSION:** So we can conclude that  $\Delta Q(\alpha) > 0, \alpha \in [0, 1]$  holds when  $l < 100\mu m$ . When  $l > 100\mu m$  and  $C_L$  is small,  $\Delta Q(\alpha) < 0$ , which makes case(b) better. **So, with min-size devices, we can keep optimum by only allowing high-Vdd buffers drive low-Vdd ones, when largest wire length of unbuffered interconnect is less than 100 $\mu m$ .**

Note: The limitation of my current results is due to the insufficiency of testing parameters. So the future work is to collect more parameters of different buffer sizes under 130nm, 90nm and 65nm.

### 3 Work3: Summarization of Weiping Shi's fast buffer insertion works

Prof. Shi's DAC03 paper is focused on the speed up of buffer insertion beyond the Van Ginniken's work (ISCAS90). The main contributions are summarized as follows,

- A. The implicit representation of tuple (RAT, capacitance), which makes the updating time  $O(1)$ .
- B. Prediction pruning, which considers pre-buffer slack, and prunes more redundancy.
- C. An fast redundancy check and merging scheme.

His ASPDAC04 paper is focused on the complexity analysis (this paper proved that cost minimization buffer insertion problem is NP-Completed) and simply extend his DAC03 work to handle triple (cost, RAT, capacitance) candidates. Actually, this approach is much like the Lillis's ICCAD05 work, in which a sorted list for cost is maintained, and each cost node points to a tuple (RAT, capacitance) candidates tree.

His ASPDAC05 paper proposes the following approximate approaches for buffer insertion,

- A. Aggressive pruning.
- B. Squeeze pruning.
- C. Buffer library lookup.

My work can follow the main idea in his DAC03 paper, and add power issue into consideration with dual-Vdd buffers. So I need to handle triple (power, RAT, capacitance) candidates. Though Shi's ASPDAC04 paper has considered a similar problem, the properties of "power" field wasn't studied carefully. I try to integrate power issue into the efficient data structures in his DAC03 paper.

In the next week, I'll focus on the extension of Shi's data structure to add power into consideration.