LETTER *Special Section on VLSI Design and CAD Algorithms*

Transistor Sizing of LCD Driver Circuit for Technology Migration

Masanori HASHIMOTO[†a)], ***Member***, **Takahito IJICHI**[†], ***Nonmember***, **Shingo TAKAHASHI**[††], ***Affiliate Member***, **Shuji TSUKIYAMA**[††], ***Member***, ***and*** **Isao SHIRAKAWA**[†††], ***Fellow, Honorary Member***

SUMMARY Design automation of LCD driver circuits is not sophisticatedly established. Display fineness of an LCD panel depends on a performance metric, ratio of pixel voltage to video voltage (RPV). However, there are several other important metrics, such as area, and the best circuit cannot be decided uniquely. This paper proposes a design automation technique for a LCD column driver to provide several circuit design results with different performance so that designers can select an appropriate design among them. The proposed technique is evaluated with an actual design data, and experimental results show that the proposed method successfully performs technology migration by transistor sizing. Also, the proposed technique is experimentally verified from points of solution quality and computational time.

key words: *technology migration, transistor sizing, LCD driver circuit*

1. Introduction

LCD driver circuits are still designed by skilled engineers manually, and CAD environment tailored for LCD panel has not been constructed. LCD panels are often manufactured in different factories even for the same product without design modification due to increase of shipment. Transistor characteristics are different in each manufacturing factory. When we manufacture LCD panels in a new factory that are compatible or superior in performance to those manufactured in other factories, a new design from scratch is too expensive in design cost. Circuit tuning of the originally designed circuit for compensating the difference of transistor parameters, such as threshold voltage and mobility, is a reasonable approach and highly demanded.

In LCD design, display fineness is commonly evaluated as a metric, ratio of pixel voltage to video voltage (RPV). The LCD driver circuit charges each pixel to the voltage of a given input video signal. There are several important performance metrics, such as RPV and area, and there is a trade-off among the metrics. Therefore, circuit designers demand a design automation technique that can analyze the trade-off.

This paper proposes a technology migration technique to provide several circuit design results with different performance so that designers can select an appropriate design considering the trade-off relation. The proposed technique adjusts transistor sizes of sampling buffer and switch in a column driver that are influential to RPV so as to compensate transistor characteristic difference. Experimental results show that technology migration succeeded with the proposed technique.

2. LCD Column Driver Circuit

2.1 Overview of LCD Driver Circuit

Figure 1 shows an overview of LCD and driver circuit. LCD driver circuits consist of a column driver circuit (Fig. 2) and a row driver circuit, and the column driver circuit is discussed in this paper, since its operation speed is much faster and it affects the overall LCD performance. In the column driver circuit, timing pulses to sampling switches are generated to capture video signal. Figure 2 explains input/output and sub-circuits. Adjacent shift registers output pulses whose timing is different by a half system clock cycle, and these sampling pulses propagate through delay and sampling buffers and finally reach to sampling switches (Fig. 3). Sampling switches get successively on and off according to the given sampling pulses. Gate line and common electrode line are stable during the sampling operation.

The delay buffer adjusts the pulse width not to overlap successive sampling pulses. The sampling buffer (Fig. 4) drives many sampling switches, and hence the load capacitance is large. Therefore, sampling buffer should be a kind of cascaded driver[1].

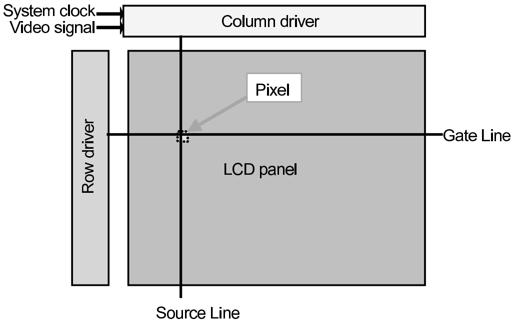


Fig. 1 LCD panel and driver circuits.

Manuscript received March 8, 2007.
Manuscript revised June 11, 2007.
Final manuscript received August 1, 2007.

[†]The authors are with the Department of Information Systems Engineering, Osaka University, Suita-shi, 565-0871 Japan.
[††]The authors are with the Department of Electrical, Electronic and Communication Engineering, Chuo University, Tokyo, 112-8551 Japan.
[†††]The author is with the Graduate School of Applied Informatics, University of Hyogo, Kobe-shi, 650-0044 Japan.
a) E-mail: hasimoto@ist.osaka-u.ac.jp

DOI: 10.1093/ietfec/e90–a.12.2712

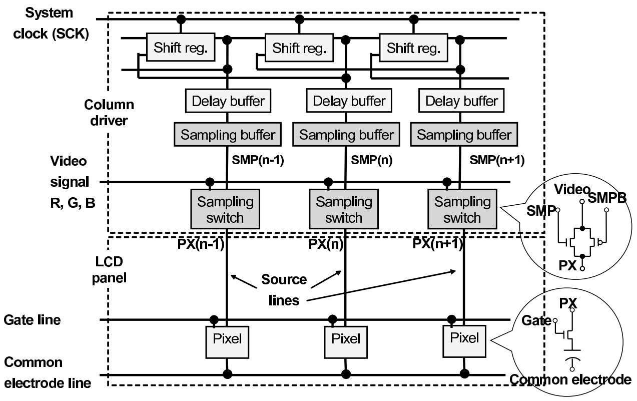


Fig. 2 LCD column driver circuit.

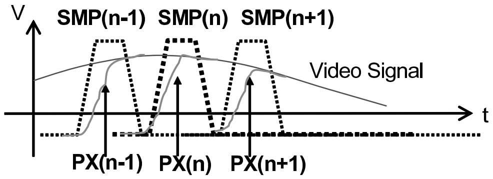


Fig. 3 SMP pulses of adjacent drivers and video signal.

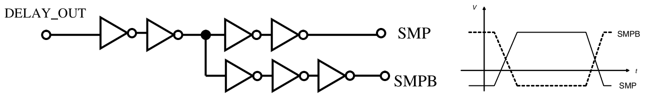


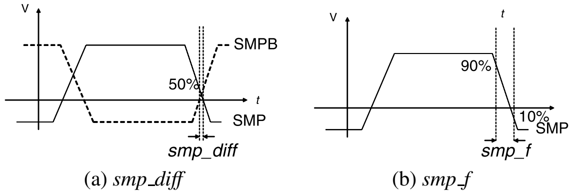
Fig. 4 Sampling buffer circuit and SMP waveform.

In designing LCD driver circuit, the sampling buffer and the sampling switch are key components to RPV performance, and are focused in this paper. Here, given a pixel, RPV is defined as RPV= $(V_{px}/V_{vd}) \times 100[\%]$, where V_{vd} denotes the video voltage to be fed to the pixel, and V_{px} is the pixel voltage after the sampling switch turns off. RPV can be below and above 100 due to incomplete feeding and charge injection, and |RPV-100| should be close to 0. The sampling buffer in Fig. 4 generates SMP and SMPB pulses whose shape should be perfectly complementary for PMOS and NMOS of sampling switches. When SMP is high, the sampling switch is on, and video signals are conducted to pixels.

2.2 Performance Metrics and Design Constraints

A primary design goal is to approach |RPV-100| to 0 under process and environmental variability. Smaller area and lower power dissipation of the designed column driver circuit are desirable.

Figure 5 shows waveform parameters *smp_diff* and *smp_f*, and they are given as a critical design constraint of SMP waveform with respect to RPV. *smp_diff* is the crossing timing difference between SMP fall and SMPB rise transitions at 50% point. *smp_f* is the 90–10% transition time of SMP. Both *smp_diff* and *smp_f* should be minimized so as to capture video signal at the accurate timing.



(a) *smp_diff* (b) *smp_f*

Fig. 5 Waveform parameters of SMP pulse.

The above performance metrics (RPV and power dissipation) and constraints (*smp_diff* and *smp_f*) must be evaluated under process fluctuation and supply voltage variation described in a specification. In addition, RPV should be evaluated varying V_{video} as well in the range given in the specification, because the charging/discharging time of a pixel through the sampling switch changes according to V_{video}. Moreover, the load capacitance of the sampling buffer depends on V_{video} because of non-linearity of transistor gate capacitance. We hence should consider the range of V_{video}.

2.3 Difficulty in Column Driver Circuit Design

We here discuss difficulties in column driver circuit design. *RPV* is dependent on sampling switch design and SMP waveform, and hence transistor widths of sampling switch and its are design parameters to determine. There are two design parameters for sampling switch (Fig. 2) and fourteen parameters for buffer (Fig. 4).

RPV evaluation must be performed under all variability conditions and voltage range of V_{dd}, V_{ss}, video, common electrode line, and hence it requires long CPU time. Thus, it is not practical to perform RPV evaluation for every transistor size change when optimizing transistor widths of sampling switch and driver. We therefore need to divide the design problem into sub-problems to reduce computational cost.

There is a trade-off between SMP waveform parameters and circuit area. For example, when enlarging sampling buffer transistors, *smp_f* is reduced and helps to improve RPV, but the circuit area increases. We have to find an appropriate point in the trade-off relation.

When evaluating SMP waveform, we have two choices; circuit simulation and gate delay model, e.g. [2]. In sampling buffer design, the important waveform parameter *smp_diff* is the delay difference of two paths, and its value is much smaller than the entire delay of the driver circuit. When gate delay estimation is not accurate, the estimation error dominates *smp_diff* value, and circuit optimization fails. Reference [2] reports that gate delay can be modeled in a posynomial expression whose variables are transistor sizes. Here, a posynomial expression is $f(x_1, x_2, \cdots, x_n) = \sum_{k=1}^{K} c_k x_1^{a_{1k}} x_2^{a_{2k}} \cdots x_n^{a_{nk}}$, where $c_k > 0$, a_{ik} is a coefficient of real number, and K is the number of terms. When an objective function and constraint functions are expressed in the posynomial expressions, the optimization problem can be formulated as a convex problem, which is known as geometric programming [3]. However, *smp_diff* includes sub-

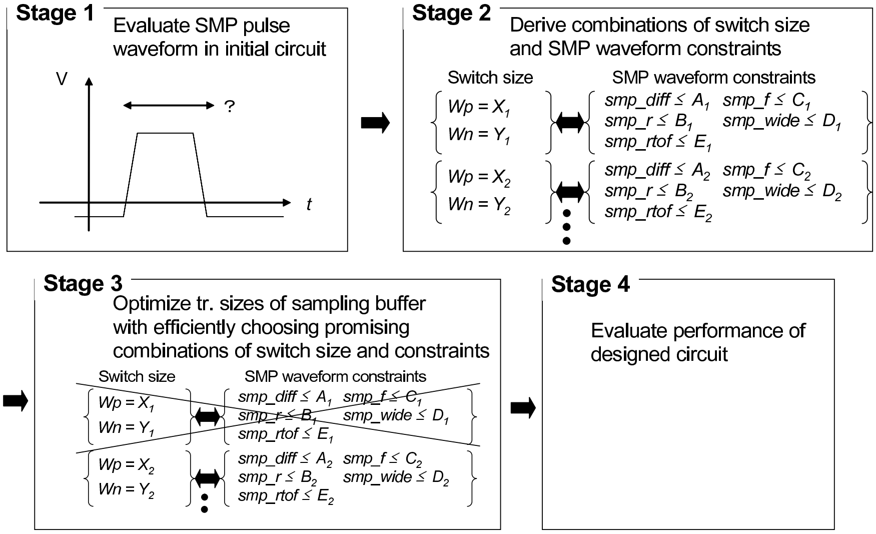


Fig. 6 Overview of proposed method. In Stage 3, non-promising combinations of switch size and SMP waveform constraints are x-ed out.

traction and it makes impossible to express *smp_diff* in a posynomial manner, because c_k should be positive. We thus have to solve a non-linear problem that may involve many local optima.

3. Proposed Method

3.1 Overview

To overcome the difficulties discussed in Sect. 2.3, we propose a design flow that divides the design problem into two sub problems; sampling switch design and sampling buffer design. The size of the design problem is decreased, and hence we can expect CPU time reduction. Figure 6 presents the proposed design flow.

Stage 1 Evaluate sampling pulse waveform of a given initial circuit.

Stage 2 Derive switch size under a given SMP waveform constraint such that |RPV-100| is minimized. We list combinations of SMP waveform constraint and switch size with extensive circuit simulations and binary search of parameters [4]. The SMP waveform constraint is set based on the evaluation results in Stage 1.

Stage 3 Design sampling buffer. During the design, non-promising combinations derived in Stage 2 are discarded by solution pruning for computational time reduction.

Stage 4 Evaluate performance of the designed circuit.

In the proposed flow, a definition of parameters for describing SMP waveform is necessary. Three parameters *smp_r, smp_wide* and *smp_rtof* shown in Fig. 7 are defined in addition to *smp_diff* and *smp_f* in Fig. 5.

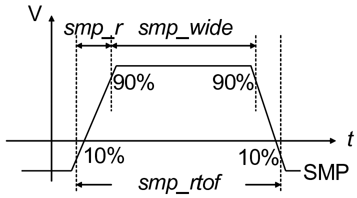


Fig. 7 Parameters of SMP waveform.

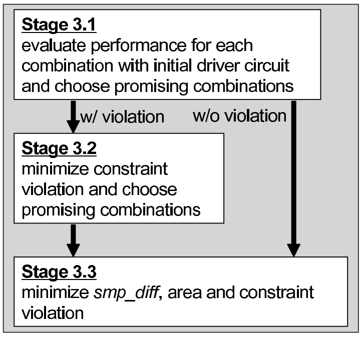

Stage 3.1
evaluate performance for each combination with initial driver circuit and choose promising combinations
w/ violation
w/o violation
Stage 3.2
minimize constraint violation and choose promising combinations
Stage 3.3
minimize smp_diff, area and constraint violation

Fig. 8 Design process of sampling buffer circuit.

3.2 Sampling Buffer Design

We here explain the proposed method of sampling buffer design in Stage 3.

3.2.1 Optimization Flow

Figure 8 shows the optimization flow of sampling buffer consisting of three steps. Stage 3.1 evaluates the performance of the given initial sampling buffer for every combination of switch size and SMP waveform constraint. We eliminate non-promising combinations before the sampling buffer optimization. If there is a violation of SMP waveform constraint, we quickly optimize the buffer with a nonlinear optimizer such that the constraint violation is minimized by sizing transistor widths in the sampling buffer in Stage 3.2. This optimization aims to estimate transistor size increase needed to reduce violation. Promising combinations are selected for Stage 3.3. Stage 3.3 performs transistor sizing to minimize *smp_diff*, *smp_size_sum* and constraint violation. Here, *smp_size_sum* is the total gate width of transistors in the sampling buffer.

3.2.2 Objective Function

We explain objective functions obj used in Stage 3.2 and Stage 3.3.

$$obj = excess\_sum \quad (\text{Stage 3.2}) \tag{1}$$

$$obj = obj\_min + excess\_sum \quad (\text{Stage 3.3}) \tag{2}$$

Stage 3.2 minimizes constraint violation and Stage 3.3 optimizes *smp_diff*, *smp_size_sum* and constraint violation with a weighting parameter. A rough optimization in Stage 3.2 tells how large transistors are necessary for violation elimination. This information is used for solution pruning before the detailed optimization in Stage 3.3, which will be explained in Sect. 3.2.3.

Function *excess_sum* is to compute the sum of constraint violation of SMP waveform and circuit area, and is expressed as follows.

$$\begin{aligned} excess\_sum =\ & excess(smp\_diff) + excess(smp\_r) \\ & + excess(smp\_f) + excess(smp\_wide) \\ & + excess(smp\_rtof) \\ & + A \cdot excess(smp\_size\_sum), \end{aligned} \tag{3}$$

where A is a coefficient to balance violations of SMP waveform constraint and circuit area constraint. $excess$ is a function to return constraint violation value, and it returns zero when the constraint is satisfied. Here, while we assign the same weighting coefficient of 1 for simplicity to $excess(smp\_diff)$, $excess(smp\_r)$, $excess(smp\_f)$, $excess(smp\_wide)$ and $excess(smp\_rtof)$, different values could be assigned if necessary.

Function *obj_min* is expressed as a sum of normalized *smp_diff* and *smp_size_sum* with a weighting coefficient *size_rate*.

$$\begin{aligned} obj\_min =\ & \frac{smp\_diff}{smp\_diff\_u}(1 - size\_rate) \\ & + \frac{smp\_size\_sum}{smp\_size\_sum\_u} \cdot size\_rate, \end{aligned} \tag{4}$$

where $0 \leq size\_rate \leq 1$. smp_diff_u and $smp\_size\_sum\_u$ are parameters for normalization. The coefficient *size_rate* enables trade-off analysis between *smp_diff* and *smp_size_sum*, and is set by designers according to the design goal. Here, other metrics, such as power consumption, could be added to the objective function.

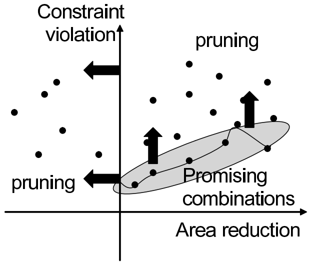


Fig. 9 Plots between constraint violation and area reduction for solution pruning in Stage 3.1.

3.2.3 Solution Pruning

To explore solution space efficiently, we execute solution pruning in Stage 3.1 and Stage 3.2. This subsection explains policies of solution pruning. Intensity of solution pruning affects final solution quality and CPU time. When using the proposed technique, we appropriately choose the degree of pruning following the policies below according to the given design time.

1. Smaller switch size desirable
   A sampling buffer drives a number of switches, and hence switch size has a large impact on circuit area and power consumption. Larger switch size also makes buffer design difficult because large load degrades design freedom.
2. Smaller constraint violation preferable
   Larger violation requires more effort, such as area and power, to fix violation.
3. Various switch sizes eligible
   Similar switch sizes result in similar design results, and explored solution space is limited.

Promising combinations of switch size and SMP waveform constraint are selected and poor combinations are discarded, for example, in the following procedure in Stage 3.1. Figure 9 plots constraint violation *excess_sum* versus switch size reduction *switch_size_reduction*, and each dot corresponds to a combination of switch size and SMP waveform constraint, where *switch_size_reduction* is the total gate width reduction of p- and n-transistors in all sampling switches connected with a single source line compared to the given initial circuit. Combinations in the circle are good in the trade-off between violation and switch size, and should be selected.

4. Experimental Results

We implemented the proposed technique with a circuit simulator [5] for performance evaluation, and numerical optimization is performed with Sequential Quadratic Programming[6], because accurate timing evaluation, especially for *smp_diff*, is necessary, and the problem has numerous local optima.

An initial circuit to optimize for technology migration

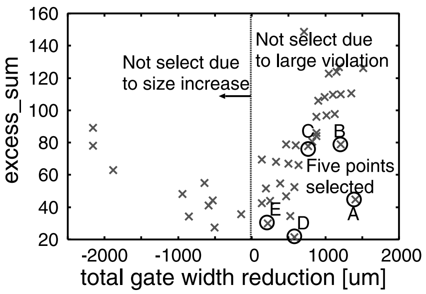


Fig. 10 Results of Stage 3.1 and selected combinations.

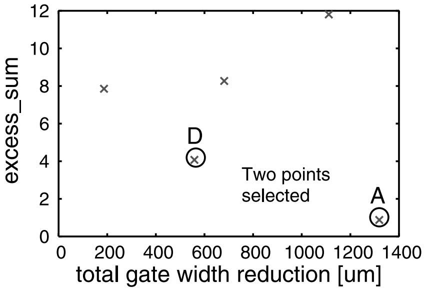


Fig. 11 Results of Stage 3.2 and selected combinations.

Table 1 Optimization results in Stage 3.3.

	A	D
excess_sum	0	2.25
total gate width reduction (μm)	1317.9	473.8

is given. In this experiment, it is assumed that threshold voltage and mobility are varied in the target technology. In Stage 1, we first evaluate the initial circuit performance and give the information to sampling switch design in Stage 2. In this experiment, Stage 2 generates 45 combinations with the procedure reported in Ref. [4].

Stage 3.1 evaluates SMP waveform with the initial sampling buffer. To efficiently save CPU time, we choose promising combinations. Figure 10 plots constraint violation for every combination. According to the selection policy discussed in Section 3.2.3, five combinations (A-E) are selected. The required CPU time is two hours.

Stage 3.2 minimizes constraint violation for the combinations selected in Stage 3.1. The optimization result is shown in Fig. 11. Two combinations (A, D) are selected; one with small constraint violation and the other is large area reduction. Stage 3.2 needs fourteen hours.

Table 1 lists the optimization results in Stage 3.3. *size_rate* is set to 1. In the case of combination (A), the violation disappears and the total circuit area is reduced by 1318μm. As for combination (D), a slight violation remains, and the area reduction is 474μm. We hence choose combination (A) as the best solution. Stage 3.3 takes ten hours.

Figure 12 shows the distributions of RPV before and after technology migration under process and environmental variation. The maximum of |RPV-100| becomes smaller after technology migration. We can also see that the number of samples whose RPV is close to 100 increases, and the performance is improved.

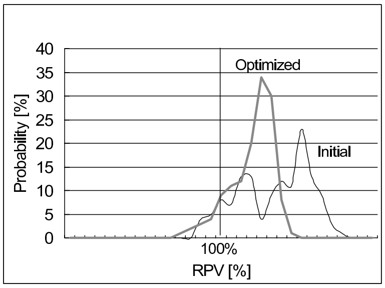


Fig. 12 RPV distributions before and after technology migration.

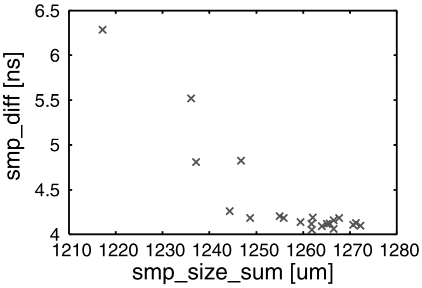


Fig. 13 Trade-off between *smp_diff* and *smp_size_sum*.

We evaluate the appropriateness of solution pruning. We executed Stage 3.2 and Stage 3.3 for the same 45 combinations, and found that combination (A) achieves the largest area reduction in the combinations with zero constraint violation, which means the solution pruning does not degrade the solution quality in this experiment. We also evaluated the CPU time reduction by solution pruning. Compared with optimization without pruning, the CPU time is reduced from 330 hours to 26 hours by over 90%.

We finally demonstrate the trade-off between *smp_diff* and *smp_size_sum*. We executed optimization in Stage 3.3 varying *size_rate*. Figure 13 shows the result, and we can see that selection of *size_rate* changes the performance as we expected.

5. Conclusion

This paper proposes a technology migration technique for LCD driver circuits. To efficiently perform transistor sizing, we divide the optimization problem into two sub-problems. Sampling switch design first derives combinations of switch size and SMP waveform constraint. In the following, sampling buffer is optimized for the given combinations. To reduce computational time, solution pruning based on several policies is executed in sampling buffer design process. Experimental results show that the proposed technique successfully executed technology migration. We also confirm that the problem division and solution pruning enable sampling buffer design in a practical time.

References

[1] H.B. Bakoglu, Circuits, Interconnections, and Packaging for VLSI, Addison-Wesley Publishing Company, 1990.

[2] J.P. Fishburn and A.E. Dunlop, “TILOS: A posynomial programming approach to transistor sizing,” Proc. ICCAD, pp.326–328, 1985.

[3] S. Boyed and L. Vandenberghe, Convex Optimization, Cambridge University Press, 2004.

[4] S. Takahashi, S. Tsukiyama, M. Hashimoto, and I. Shirakawa, “A sampling switch design for liquid crystal displays,” Proc. IEEE Int’l Region 10 Conf., no.1C-03.3, 2005.

[5] Silvaco Inc., SmartSpice User’s Manual, 2005.

[6] C. Laurence, J.L. Zhou, and A.L. Tits, User’s Guide for CFSQP Version 2.5: A C Code for Solving (Large Scale) Constrained Nonlinear (Minimax) Optimization Problems, Generating Iterates Satisfying All Inequality Constraints, April 1997.