An Optimization Technique For Dual-Output Domino Logic

Sumant Ramprasad, Ibrahim N. Hajj, and Farid N. Najm
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign, Urbana IL, USA 61801.

1 Introduction

Dynamic logic circuits [2] are used in high-performance circuits due to their speed and area advantage over static CMOS circuits. One well-known dynamic logic family is the domino CMOS family, which, however, suffers from its inability to perform inversions. Various methods have been proposed to overcome this restriction. One such method is the dual-output domino logic family. In the standard dual-output domino logic gate shown in Figure 1 each dual-output gate consists of two standard domino logic gates, producing the output, R and its complement, $\overline{R}$. The advantage of the dual-output

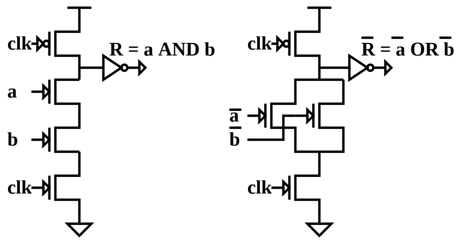

Figure 1: Standard Dual-Output Domino Logic AND2 Gate

domino CMOS family is its completeness, but at the cost of higher area and higher power dissipation. In a combinational logic block implemented using domino CMOS, only the fanin cones of inverters have to be dual-output. In this paper, by dual-output domino CMOS, we mean that only those gates whose both outputs are needed are dual-output. For instance, in the circuit in Figure 2, taken from [1], only $G7$, $G3$, $G8$, $G12$, $G13$, and $G14$ have to be dual-output. The remaining gates are single output.

In the clock-delayed (CD) domino logic style [3], the clock to a gate is delayed using delay elements attached to a gate till the inputs are known to have reached their final value. The general CD domino scheme is faster than static CMOS but typically consumes more area. The disadvantage of CD domino is that delay elements are required to be attached to gates. CD domino logic may also be slower than dual-output domino since a margin is added to the delay elements by making the delay element slower than the gate it is attached to. One method to overcome this problem is to use dual-output domino for the critical path and CD domino logic for gates

This work was supported by NSF award MIP-97-10235.

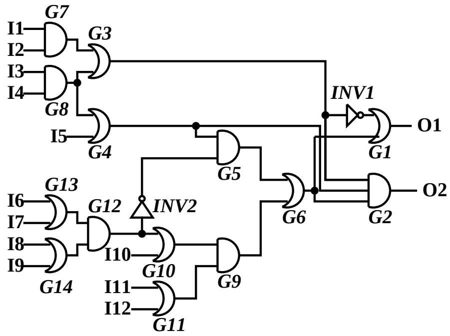

Figure 2: Example circuit

not on the critical path. In our paper, this idea is further extended in the clock-generating (CG) domino logic scheme by recognizing that a delayed clock can be generated for gates not on the critical path from the dual-output gates that are on the critical path. This eliminates the delay elements required by CD domino. CG domino reduces area, power dissipation, and clock load of dual-output domino CMOS without increasing the delay. Simulation results with ISCAS 85 benchmark circuits indicate an average reduction in area, power, and clock load of 17%, 24%, and 20% respectively over dual-output domino. The performance results are typically better for larger circuits with a 48% power reduction for the largest circuit.

2 Clock-generating (CG) domino

In CG domino, given a circuit, a dual-output gate Q like the one shown in Figure 1 is first located. It is then replaced with the modified dual-output domino logic gate Q shown in Figure 3, in which there is one gate to produce the output Q. The complemented output, $\overline{Q}$, is generated by inverting

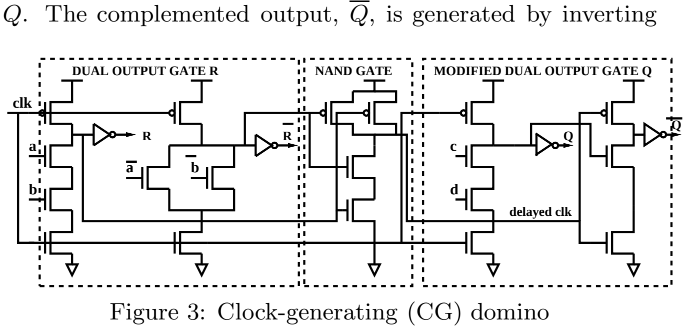

Figure 3: Clock-generating (CG) domino

Q and using it as input to an AND1 (i.e., buffer) domino gate clocked by a delayed clock. The delayed clock is a delayed version of the precharge evaluate clock and is 1 only after Q has evaluated. If $\overline{Q}$ has low fanout (1 or 2), then the buffer in Figure 3 can be eliminated by using the delayed clock for each fanout gate, P, of $\overline{Q}$ as shown in Figure 4. Figure 4 is not used if $\overline{Q}$ has high fanout because then the fanout of the delayed clock would be increased.

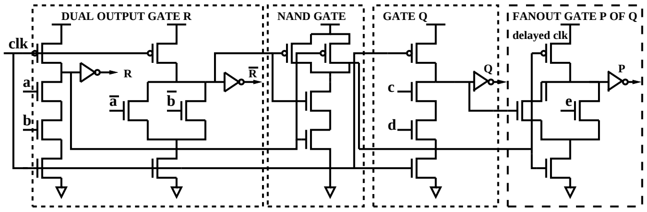

Figure 4: CG domino when gate $\overline{Q}$ has low fanout

The modified gates in Figure 3 and Figure 4 have several advantages over the gate in Figure 1, First, since fewer transistors are needed, the area and power dissipation are reduced. Second, since the complemented inputs $\overline{a}$, $\overline{b}$, ... are not needed, the gates in the fanin cone does not have to be dual-output. Third, it is possible to choose whether to implement Q or $\overline{Q}$ depending on their effect on power, area, and delay.

The generation of the delayed clock is discussed next.

2.1 Generation of the delayed clock

A gate has evaluated if and only if (R OR $\overline{R}$) is 1. To generate the delayed clock for a gate Q, the circuit is analyzed to locate another dual-output gate R in the circuit satisfying 4 conditions listed later in this sub-section. The delayed clock is then derived by ORing the outputs of R (or, equivalently, NANDing the outputs of R before the inverters). The generation of the delayed clock is shown in Figure 3. For a dual-output gate R to supply a delayed clock gate to another dual-output gate Q, it must satisfy the following 4 conditions. 1) There should not be a path from $\overline{Q}$ to R or $\overline{R}$. This condition is to ensure that a cycle is not introduced. 2) The delayed clock generated from R and $\overline{R}$ must arrive after the positive output of Q (we included a margin of 10% to be conservative). This condition ensures that the delayed clock arrives after the positive output of Q and hence there is at most one rising transition at $\overline{Q}$. 3) The delayed clock generated from R and $\overline{R}$ must arrive before the latest required time for any input to the gate generating $\overline{Q}$ (we, again, included a margin of 10% to be conservative). This condition ensures that there is no increase in the delay through the combinational block. 4) The gate R must not be supplying the clock to more than a certain number of gates (this limit was set at 8). This condition is to avoid violating any fanout constraints.

Note that the modified dual-output gate Q in Figure 3 can also be used to generate the delayed clock signal for other gates. It is, however, not possible to use the gate Q in Figure 4 to generate a delayed clock.

2.2 Example

In this sub-section, the circuit in Figure 2 is used as an example to illustrate CG domino. The inverters $INV1$ and $INV2$ cannot be propagated towards the primary inputs, due to which gates $G3$, $G7$, $G8$, $G12$, $G13$, and $G14$ have to be dual-output. Assuming all the primary inputs arrive at the same time, $G12$ can supply a delayed clock to $G3$ since it can be shown that $G12$ and $G3$ satisfy the 4 conditions listed in subsection 2.1. Since $\overline{G3}$ has a fanout of 1, the optimization in Figure 4 is used due to which $G3$, $G7$, and $G8$ no longer have to be dual-output and can now be single output. The delayed clock obtained by NANDing the outputs of $G12$ before the inverters is used as the clock for $G1$. Layouts of the circuit in Figure 2 using standard dual-output domino and CG domino are shown in Figure 5 and Figure 6 respectively. A SPICE

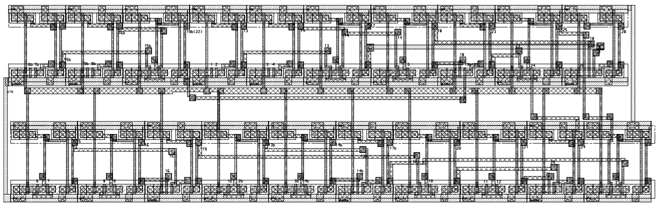

Figure 5: Circuit layout using standard dual-output domino

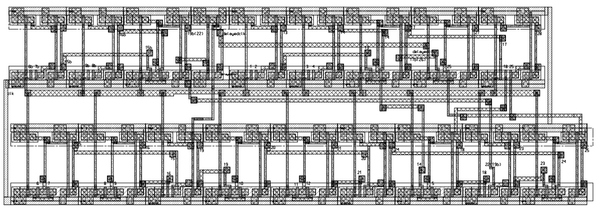

Figure 6: Circuit layout using CG domino

file was extracted from both layouts and simulated. In the CG domino circuit, the delay of the path $G7$-$G3$-$INV1$-$G1$ is increased from 2.62ns to 2.90ns. This increase does not, however, slow down the circuit since it is less than the critical path delay of 5.7ns through the critical path $G13$-$G12$-$G10$-$G9$-$G6$-$G2$. The critical path is unchanged in the CG domino circuit and hence the critical path delays are the same in the two circuits. The number of transistors is reduced from 121 to 107, the power dissipation is reduced from 0.824 mW to 0.737 mW, and the capacitance of the clock network is reduced from 399fF to 338fF.

One timing diagram from a simulation of the CG domino circuit is shown in Figure 7. When clk goes high, either $G12$ or $\overline{G12}$ is 1 after a certain time. This results in delayed clk becoming 1 due to which $G1$ evaluates. Note that one input, $\overline{G3}$, to $G1$ is a falling input, i.e., $\overline{G3}$ makes at most a single 1 to 0 transition during the evaluate phase. This is acceptable since the clock to $G1$ is delayed until after $\overline{G3}$ has reached its final value. There is a short spike on $G1$ after the second evaluate because delayed clk goes low after $\overline{G3}$ becomes 1. This spike does not cause an error because it occurs during precharge and because it is short. If needed, the spike can be eliminated by ensuring that delayed clk goes low before $\overline{G3}$ becomes 1.

3 Synthesis of CG domino

The synthesis procedure for CG domino circuits is shown in Figure 8. The input is a boolean function which is mapped to

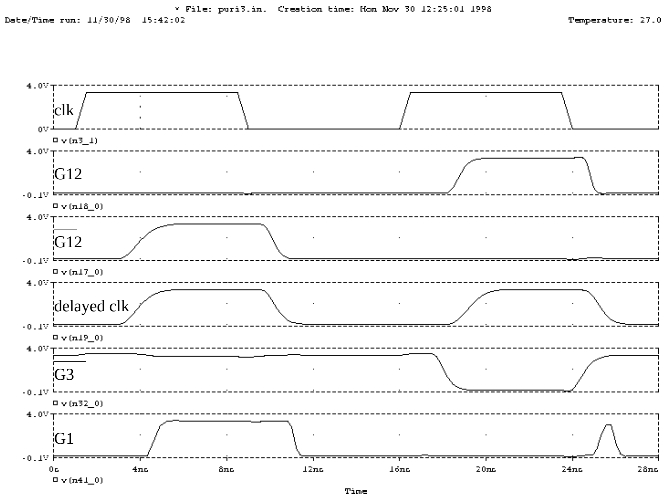

Figure 7: Timing diagram for circuit using CG domino

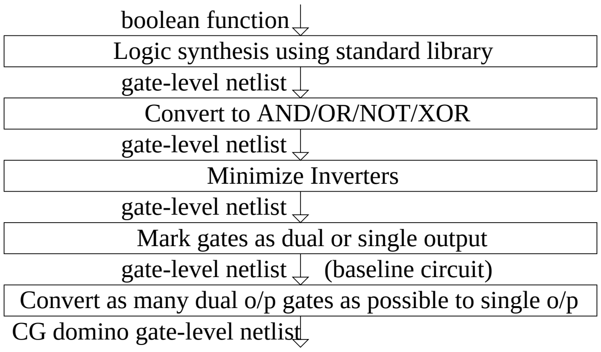

Figure 8: Synthesis of CG domino circuits

a standard library using the SIS logic synthesis tool. The output of SIS is a gate-level netlist containing both inverting and non-inverting gates. This netlist was converted to a netlist comprising of only AND, OR, NOT, and XOR2 gates. The number of NOT gates was minimized by propagating them towards the primary inputs, wherever possible. Inverters at the primary inputs and primary outputs were deleted since these inverters can be absorbed into the registers at the input and output of the combinational circuit block. At this point, we have a domino logic circuit in which only gates whose both outputs were required were duplicated. This was the baseline circuit used for comparison. The next step was to do a timing analysis on the circuit to determine the arrival times at each node. The following two optimizations were implemented on this baseline circuit based on the timing analysis. The first optimization was to use the timing information to convert some of the dual-output gates to single output. The idea behind this optimization is that a domino AND gate can accept falling inputs if there is a rising input guaranteed to arrive later. This optimization, typically, converted a small amount of gates from dual-output to single output. The second optimization was to use CG domino logic by modifying dual-output gates by finding another dual (or modified-dual) output gate that will supply a delayed clock.

Table 1: Experimental results for ISCAS 85 benchmark circuits

Circuit	Area (Transistor count)			Clock Load (Transistor count)			Power (Switching Activity)		
	Dual Output domino	CG domino	% Redn.	Dual Output domino	CG domino	% Redn.	Dual Output domino	CG domino	% Redn.
c432	1121	1086	3	328	308	3	85	86	0
c499	2192	2192	0	584	584	0	145	146	0
c880	2554	2164	15	830	660	20	144	108	25
c1355	4664	4164	11	1540	1316	15	296	240	19
c1908	4541	4281	6	1410	1288	9	313	286	9
c2670	7392	5955	19	2334	1862	20	577	404	30
c3540	11391	9179	19	3594	2748	24	889	687	23
c5315	14529	11177	23	4550	3308	27	989	631	36
c6288	21930	20948	4	7310	6934	5	2279	2192	4
c7552	24372	17403	29	7790	5244	33	1853	966	48
Total	94686	78549	17	30266	24252	20	7570	5746	24

4 Experimental Results

The synthesis procedure described in section 3 was implemented in a combination of C and Perl. The ISCAS 85 benchmark circuits were used to measure the improvement achieved using CG domino. We measured the total number of transistors, which is a measure of area, and the total number of transistors driven by the precharge-evaluate clock, which is a measure of the clock load. In addition, a zero-delay simulation was performed for 500 clocks with random input vectors to measure the average switching activity at each node. A zero-delay model was used since there are no glitches in domino logic. Assuming the capacitances of the gates are equal, the average switching activity at each node was summed up to give a measure of the total power dissipation of the entire circuit. A unit-delay model was assumed in order to estimate the arrival times since most of the gates are 2-5 input AND/OR gates with fanouts of 1-5. CG domino is not limited to using a unit-delay model and any other delay model can also be used. Our main aim in conducting the experiments was to estimate the relative benefit over dual-output domino CMOS.

The results are shown in Table 1 from which it can be seen that the results are typically better for larger circuits with many levels of logic. For instance, there is a reduction in power of 48% for the largest circuit (c7552). The optimizations took less than a minute to perform for each circuit on a Sun Sparc Ultra-1.

References

[1] R. Puri, A. Bjorksten, T. E. Rosser, "Logic Optimization by Output Phase Assignment in Dynamic Logic Synthesis," *International Conference on Computer-Aided Design*, pp. 2–7, San Jose CA, November 10–14 1996.

[2] N. Weste and K. Esraghian, "Principles of CMOS VLSI design: a systems perspective," 2nd Edition, Addison Wesley, Reading MA, 1993.

[3] G. Yee and C. Sechen, "Dynamic Logic Synthesis," *Custom Integrated Circuits Conference*, pp. 345–348, Santa Clara CA, May 1997.