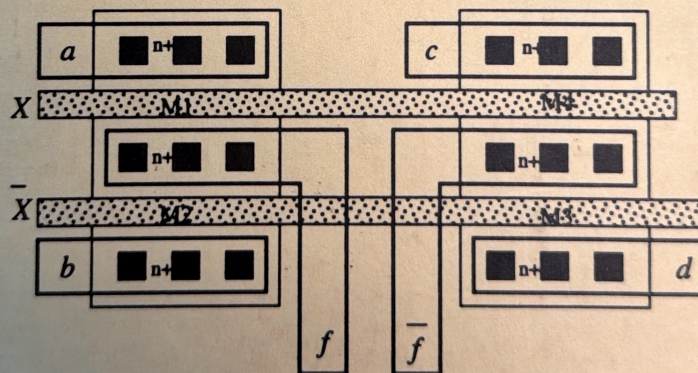
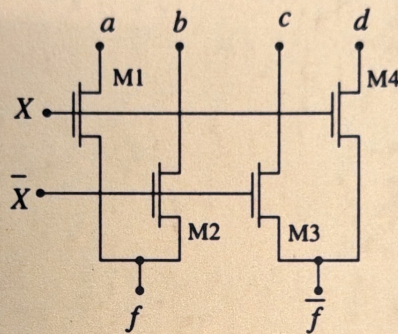


CMOS LOGIC CIRCUIT DESIGN



John P. Uyemura

CMOS LOGIC CIRCUIT DESIGN

John P. Uyemura
Georgia Institute of Technology



KLUWER ACADEMIC PUBLISHERS
Boston / Dordrecht / London

Distributors for North, Central and South America:

Kluwer Academic Publishers
101 Philip Drive
Assinippi Park
Norwell, Massachusetts 02061 USA
Telephone (781) 871-6600
Fax (781) 871-6528
E-Mail <kluwer@wkap.com>

Distributors for all other countries:

Kluwer Academic Publishers Group
Distribution Centre
Post Office Box 322
3300 AH Dordrecht, THE NETHERLANDS
Telephone 31 78 6392 392
Fax 31 78 6546 474
E-Mail <orderdept@wkap.nl>



Electronic Services <<http://www.wkap.nl>>

Library of Congress Cataloging-in-Publication Data

A C.I.P. Catalogue record for this book is available
from the Library of Congress.

Copyright © 1999 by Kluwer Academic Publishers

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher, Kluwer Academic Publishers, 101 Philip Drive, Assinippi Park, Norwell, Massachusetts 02061

Printed on acid-free paper.

Printed in the United States of America

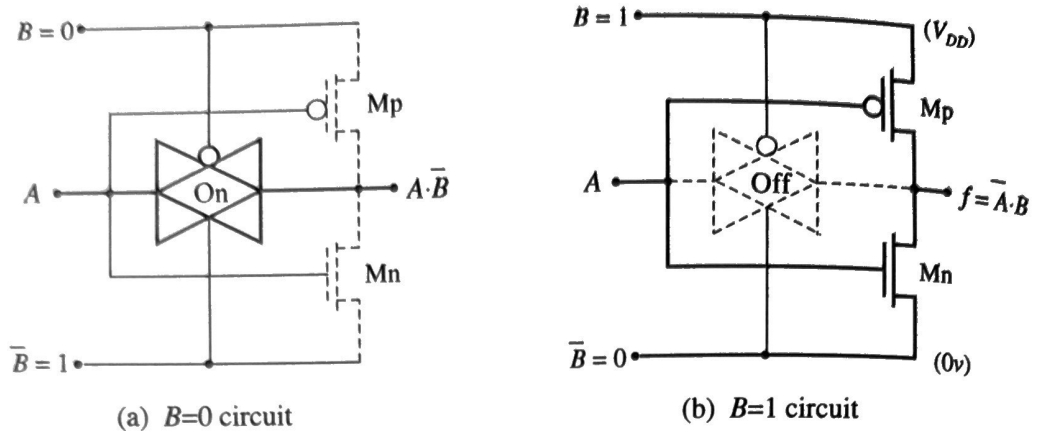


Figure 6.24 Operation of the TG XOR circuit

6.4.4 Transmission-gate Adders

Consider two logic bits a_n and b_n , and a carry-in bit c_n . Recall that the full-adder produces the sum s_n and carry-out c_{n+1} by means of

$$\begin{aligned} s_n &= a_n \oplus b_n \oplus c_n \\ c_{n+1} &= a_n b_n + c_n (a_n \oplus b_n) \end{aligned} \quad (6.43)$$

Since transmission gate logic allows for direct implementation of the XOR function, various full-adder networks can be constructed using the TG circuits discussed above.

Figure 6.24 shows a full adder that creates the sum bit s_n directly using two XOR operation expressed in

$$\begin{aligned} s_n &= a_n \oplus b_n \oplus c_n \\ &= (a_n \oplus b_n) \cdot \bar{c}_n + (\overline{a_n \oplus b_n}) \cdot c_n \end{aligned} \quad (6.44)$$

where $(a_n \oplus b_n)$ and $\overline{(a_n \oplus b_n)}$ are generated in the left circuits, and the last XOR is obtained using the two TGs in the upper right portion of the circuit. The carry out bit c_{n+1} is obtained by rewriting the expression in the form

$$c_{n+1} = (a_n \oplus b_n) \cdot c_n + \overline{(a_n \oplus b_n)} \cdot a_n \quad (6.45)$$

One unique aspect of this adder is that the outputs s_n and c_{n+1} are available at approximately the same time. This is due to the fact that the propagation delays from the inputs to the outputs are almost equal because of the symmetry used to construct the circuits.

6.5 TG Registers

Transmission gates can be used as simple switches to create circuits that have at least two distinct operational modes:

- **Load** - The value of a bit D is used as an input to the circuit, and,
- **Hold** - The input line is disconnected from the circuit, and the value is held.

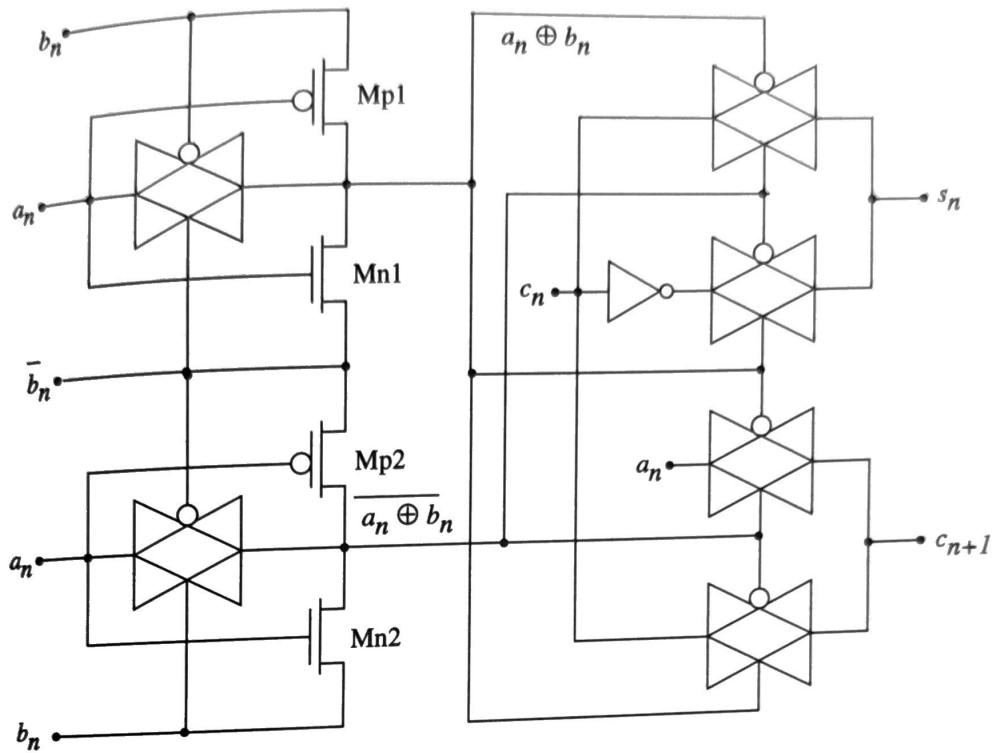


Figure 6.25 Transmission gate full-adder circuit

This allows us to create memory circuits that can be used for latches, registers, and other state elements. Figure 6.26 shows the basic circuit for a level-sensitive D latch. The operation is controlled by the load signal LD . A value of $LD = 1$ accepts the input bit D into the network. Switching the control to a value of $LD = 0$ allows the circuit to hold the value.

The operation of the latch is detailed in Figure 6.27. The load operation is shown in Figure 6.27(a). When $LD = 1$, the input transmission gate TG1 acts as a closed switch, while TG2 is open. The value of the input data bit D enters the circuit, and is available at the outputs Q and $\bar{Q}$. Since the outputs change in response to a change in D , the latch is said to be **transparent**. The hold operation

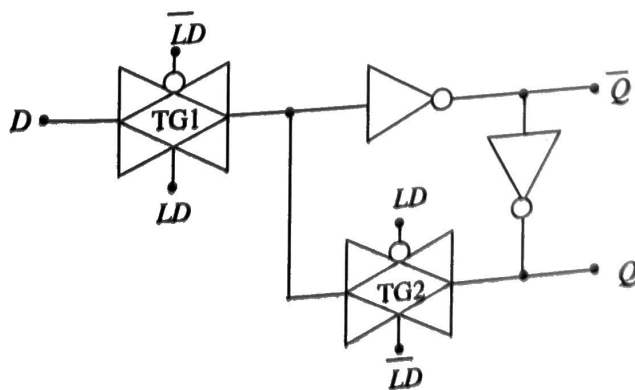
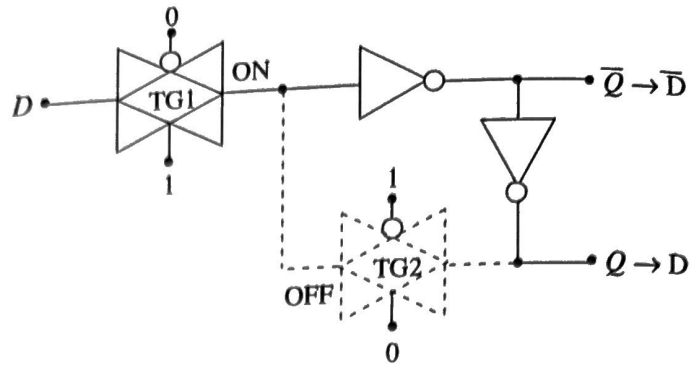
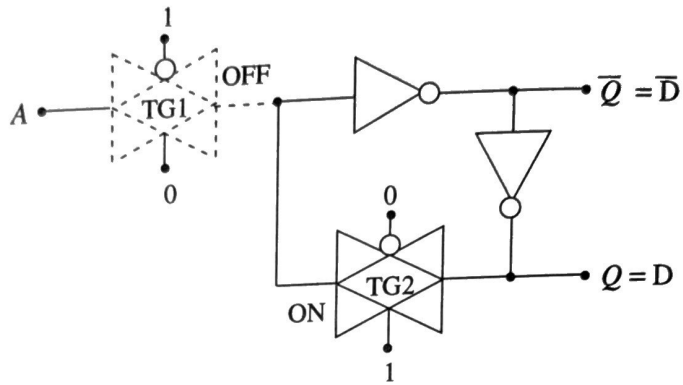


Figure 6.26 Basic TG latch circuit

(a) Load operation with $LD=1$ (b) Hold operation with $LD=0$ **Figure 6.27** Operation of the TG D-type latch

occurs when $LD=0$, giving the circuit in Figure 6.27(b). In this case, TG1 is open, disconnecting the input line from the circuit; the new input bit A cannot enter the circuit. Transmission gate TG2 is closed, and completes the feedback loop between the inverters, changing it into a bistable circuit that can store either a 0 or 1 state. The actual value in the circuit is to D that was acquired when the LD signal changed from a 1 to a 0.

6.6 The D-type Flip-Flop

Many CMOS circuits use a **clock signal** $\phi(t)$ as shown in Figure 6.28 for their operation. The clock provides a simple way to synchronize operations in a digital network relative to an absolute time base. The period of the clock is T seconds corresponding to a frequency $f=(1/T)$ Hz. The complement $\bar{\phi}(t)$ is also shown in the drawing. Since transmission gates can be switched ON or OFF with a complementary pair, the clock signal provides a method for synchronizing data flow. This is a common technique in classical CMOS design, and is a straightforward application that demonstrates the utility of a TG in timing circuits.

We will create a D flip-flop (DFF) that triggers on the positive edge by (a) cascading two oppositely phased D-latches as shown in Figure 6.29. This is a standard master-slave arrangement that allows us to create a non-transparent latching circuit. The master and slave circuits are identical, but are out of phase due to the clock signal distribution.

The operation of the circuit can be understood using the circuits shown in Figure 6.30. When

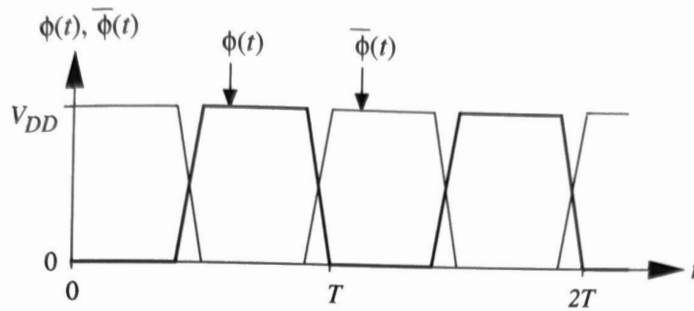


Figure 6.28 TG clocking signals

the clock is at a value of $\phi=0$ ($\bar{\phi}=1$), TG1 is in a conducting mode and passes the data input bit D to the first (master) latch as shown in Figure 6.30(a). Since TG2 and TG3 are both OPEN during this time, no further data transfer takes place. When the clock makes a transition to $\phi=1$ ($\bar{\phi}=0$) as in drawing (b), TG1 acts like an open circuit and block changes in the data. During this time, TG2 closes and completes the feedback latching circuit, while TG3 is CLOSED allowing the data voltage to be transmitted into the second (slave) latch. The output Q at this time is the value of D that was present when the clock made a transition from $\phi=0$ to $\phi=1$, making this a **positive edge triggered** storage element.¹ Figure 6.31 shows the symbol and clock diagram for the DFF. A negative-edge triggered DFF is easily constructed by just reversing ϕ and $\bar{\phi}$ everywhere in the circuit.

Logically, we describe the operation of the DFF using the expression

$$D(t) = Q(t + T) \quad (6.46)$$

This states that the present value of D becomes the next value of Q at the output; this implies that the time t when the clock changes. While this is useful for latching data for the next clock cycle, it does not provide a HOLD state where the value may be stored in the circuit. This is easily remedied using the circuit in Figure 6.32. Everything is identical except that the control signals applied to TG1 have been modified to the composite signals

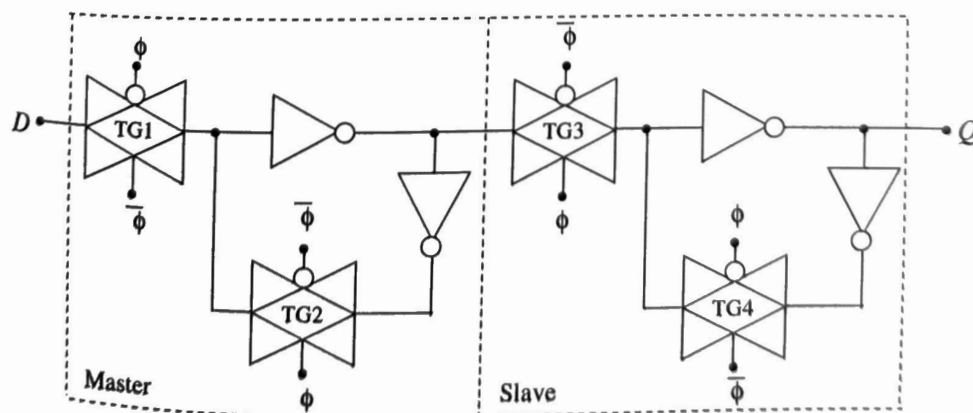


Figure 6.29 TG-based master-slave flip-flop

¹ Strictly speaking, the master-slave flip-flop is not the same as an edge-triggered flip-flop, but the terminology will be used interchangeably here.

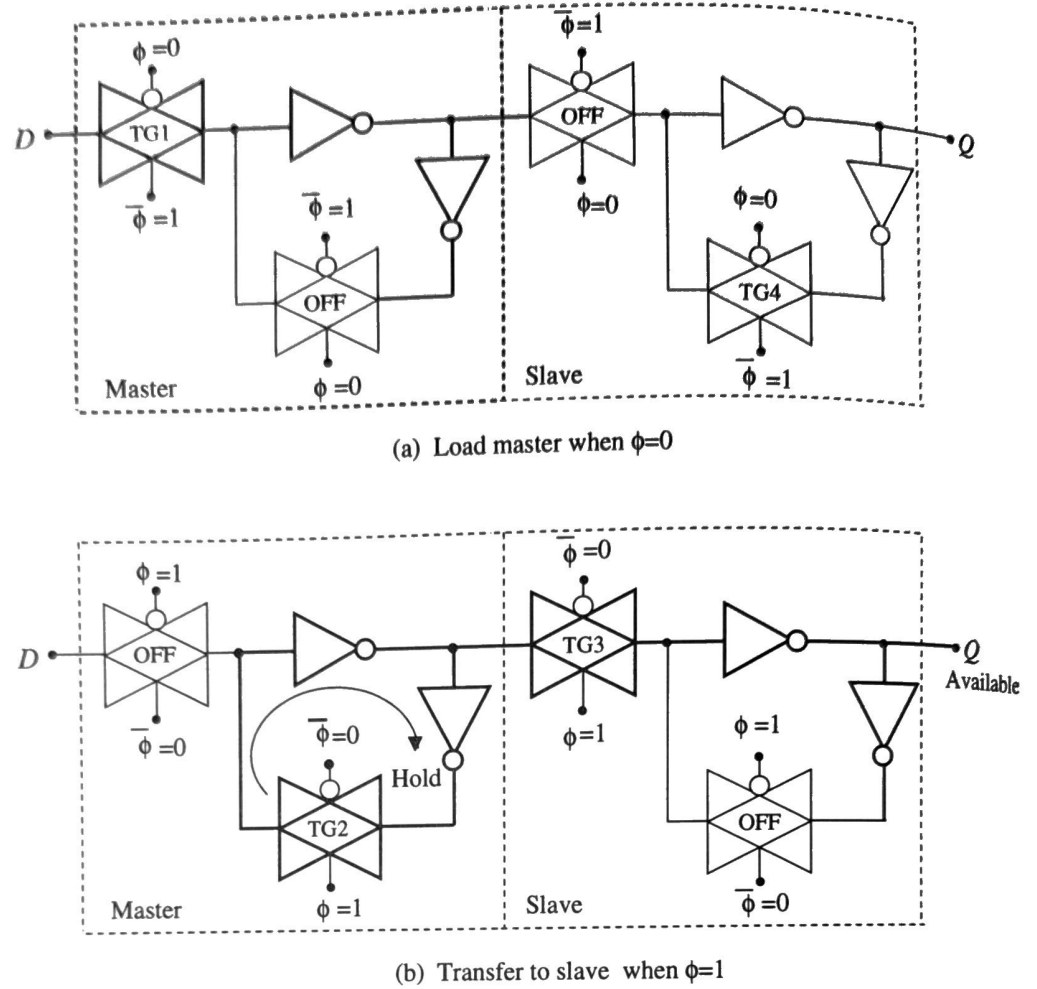


Figure 6.30 Operation of the TG flip-flop

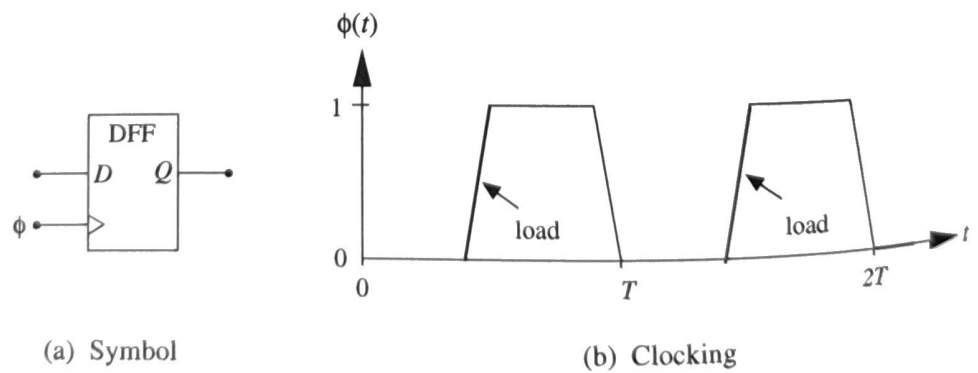


Figure 6.31 D-Flip-flop timing

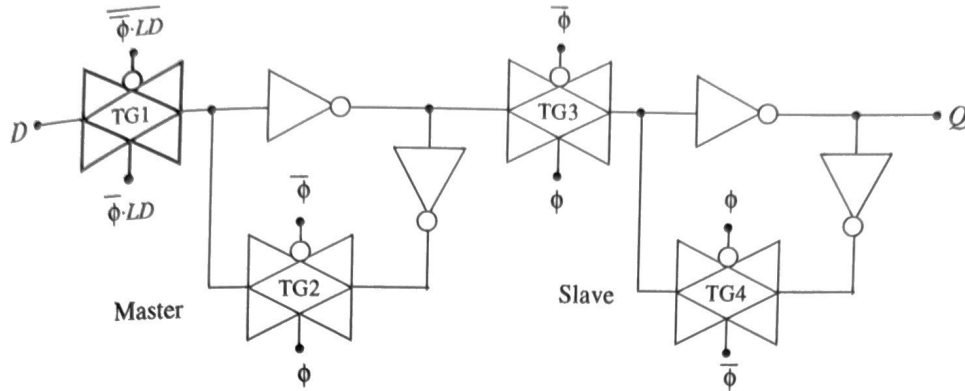


Figure 6.32 DFF with Load control

$$\bar{\phi} \cdot LD \quad \text{and} \quad \bar{\phi} \cdot \overline{LD} \quad (6.47)$$

where LD is the LOAD control. When $LD = 1$, the circuit operates as a normal DFF. However, if $LD=0$, then TG1 is kept OPEN which prohibits the entry of a new data bit into the master circuit.

6.7 nFET-Based Storage Circuits

The TG-based storage circuits above can also be constructed using only nFETs instead of transmission gates. A basic latch is shown in Figure 6.33. The input transistor M1 is controlled by the load signal LD , while M2 is controlled by $\overline{LD}$ and is used for the feedback loop in the inverter latch. The operation of the circuit is identical to the TG equivalent, and is summarized in Figure 6.34. When $LD=1$, the input data bit D is allowed into the circuit [see Figure 6.34(a)]. A value of $LD=0$ blocks the input path and simultaneously closes the feedback loop. This gives the hold condition shown in Figure 6.34(b).

Although this is simpler than the equivalent TG implementation in that it uses two less transistors (and also eliminates the additional wiring), we must be more careful at the circuit design level since the nFET only passes a limited range of voltages. In particular, we recall that applying V_{DD} to the gate allows the nFET to pass voltages in the range $[0, V_{max}]$ where

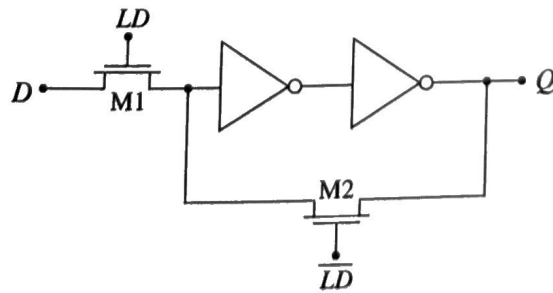


Figure 6.33 nFET-based latching circuit