overhead occurs only once. The result is improved system performance.

Listing 1 is an example of standard interrupt-handling software. Standard practice involves vectoring, pushing, and popping registers for each interrupt request. A more sophisticated function in **Listing 2** tries to handle as many interrupt requests as the peripheral requires. This function processes multiple reads to the identification register, and the process repeats until the µC has serviced all sources. You can download both listings from *EDN*'s Web site, www.ednmag.com. Click on "Search Databases" and then enter the Software Center to download the file for Design Idea #2489. (DI #2489)

Delay line eases Spice dead-time generation

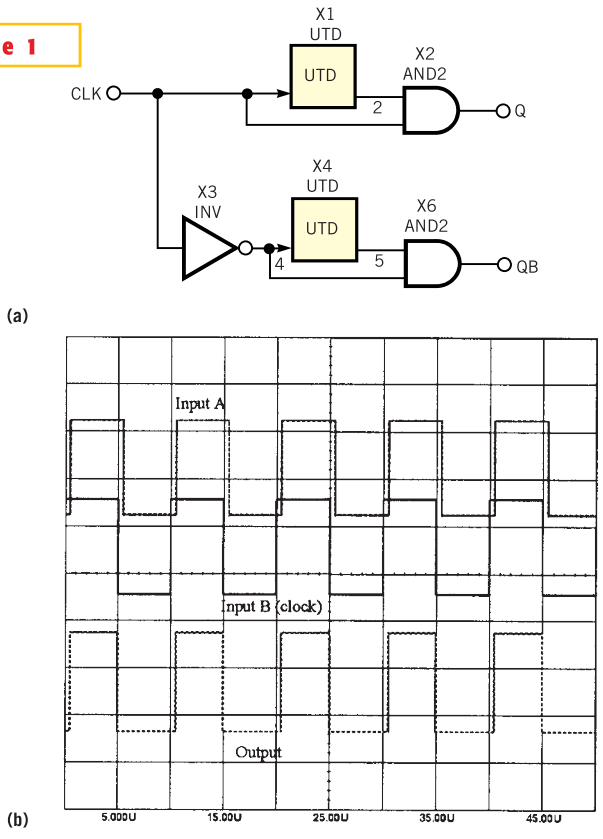
Christophe Basso, On Semiconductor, Toulouse, Cedex, France

Generating complementary clock signals in a Spice simulation is an easy task. However, this task gets much harder if you need to introduce some dead time into the signals. This difficulty is especially true when you're dealing with a variable-pulse-width-modulated switching cycle. In fact, you need to insert a dead-time interval between the switching of any two power devices in series, such as bridge or half-bridge designs that use MOSFETs and switch-mode power supplies and that implement synchronous rectification. The dead time prevents any cross-conduction, or shoot-through, between both switches and helps to reduce the associated losses.

The circuit in **Figure 1a** overcomes this typical Spice problem. The input clock drives two delay lines that feature the same specifications. When the clock goes high, one input to X2's AND gate is also high. However, because of the delay line, the other input stays low for the given dead time. When both inputs are high, the output is a logic one (**Figure 1b**).

When you generate models in a proprietary syntax, the translation process to another platform is usually painful. However, thanks to common Spice3 primitives, such as the delay line, T, the trans-

Figure 1



Two AND gates and two delay lines generate a dead-time element in Spice (a). When both inputs are high, the output is a logic one (b).

LISTING 1—HALF-BRIDGE DRIVER IN ISSPICE4

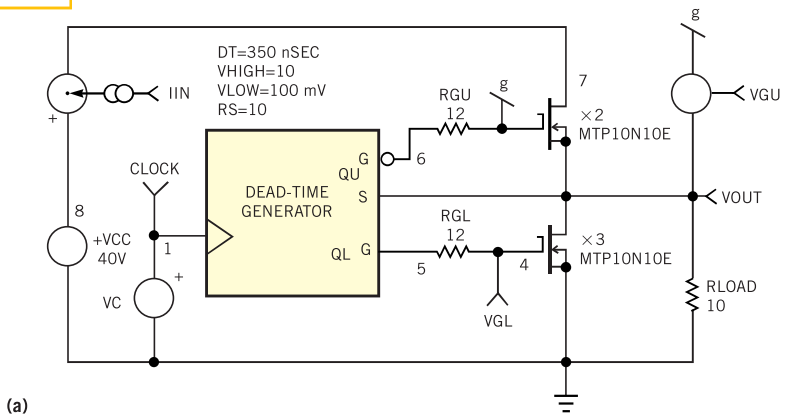
```
Listing 1 for DI #2490

.SUBCKT NEWDT CLK GU SU QL {DT=500N VHIGH=10V VLOW=100M RS=1
* Clock_In GateUpper SourceUpper GateLower
*
* DT: Dead time in seconds
* VHIGH: Output level when high
* VLOW: Output level when low
* RS: Driver's output resistance
*
BU1 1 0 V=(V(CLK)>800M) & (V(TD1)>800M) ? {VHIGH} : {VLOW}
BU2 4 SU V=V(1)
RSU 4 GU {RS}
RFLO SU 0 1G
BL 2 0 V=(V(CLKB)>800M) & (V(TD2)>800M) ? {VHIGH} : {VLOW}
RSL 2 QL {RS}
X1 CLK TD1 UTD PARAMS: TD=DT
X2 CLKB TD2 UTD PARAMS: TD=DT
X3 CLK CLKB INV
.ENDS
*INCLUDE DEAD.LIB
**********
.SUBCKT UTD 1 2 {TD=???}
*
*Parameters K=GAIN TD=DELAY
RIN 1 0 1E15
E1 3 0 1 0 1
T1 3 0 2 0 Z0=1 TD={TD}
R1 2 0 1
.ENDS
**** 1 INPUT INVERTER ****
.SUBCKT INV 1 2
B1 4 0 V= V(1)>800M ? 0 : 5V
RD 4 2 100
CD 2 0 10P
.ENDS INV
```

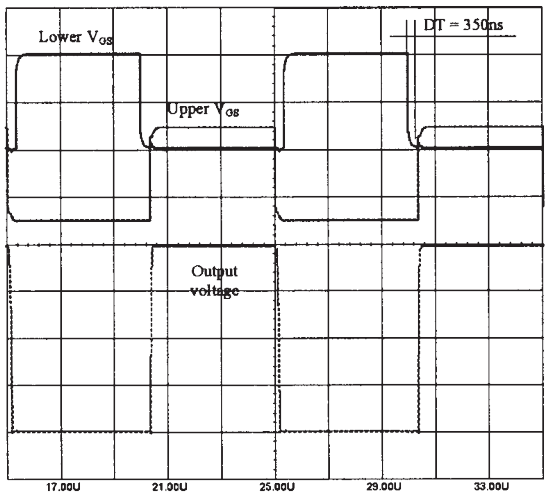
LISTING 2–HALF-BRIDGE DRIVER IN PSPICE

```
.SUBCKT NEWDT CLK GU SU QL PARAMS: DT=500N VHIGH=10V VLOW=100M
RS=10
* Clock_In GateUpper SourceUpper GateLower
*
* DT: Dead time in seconds
* VHIGH: Output level when high
* VLOW: Output level when low
* RS: Driver's output resistance
*
EBU1 1 0 VALUE = { IF ( (V(CLK)>800M) & (V(TD1)>800M), {VHIGH}, {VLOW} ) }
EBU2 4 SU VALUE = { V(1) }
RSU 4 GU {RS}
RFLO SU 0 1G
EBL 2 0 VALUE = { IF ( (V(CLKB)>800M) & (V(TD2)>800M), {VHIGH}, {VLOW} ) }
RSL 2 QL {RS}
X1 CLK TD1 DL PARAMS: TD={DT}
X2 CLKB TD2 DL PARAMS: TD={DT}
X3 CLK CLKB INV
.ENDS
**********
.SUBCKT DL 1 2 PARAMS: TD=500n
*
RIN 1 0 1E15
E1 3 0 1 0 1
T1 3 0 2 0 Z0=1 TD={TD}
R1 2 0 1
.ENDS DL
**** 1 INPUT INVERTER ****
.SUBCKT INV 1 2
EB1 4 0 VALUE = { IF ( V(1)>800M, 0, 5V ) }
RD 4 2 100
CD 2 0 10P
.ENDS INV
```

Figure 2



(a)



(b)

A typical application for the Spice dead-time generator is for simulating the operation of a half-bridge driver (a). IsSpice4 waveforms show a dead time of 350 nsec (b).

lation of this generator is easy to implement. The netlists in **listings 1** and **2** implement a half-bridge driver with a floating upper output in IsSpice4 (Intusoft) and Pspice (OrCAD), respectively. The BL (**Listing 1**) and EBL (**Listing 2**) inline equations implement the AND gates of **Figure 1a**, which saves you from using a subcircuit arrangement. Typical applications include half-bridge drivers and synchronous rectifiers. You can easily tailor any output polarity by reversing the corresponding Spice element. For instance, if you want to reverse the upper generator, BU1, in **Listing 1**, simply replace the line V=(V(CLK)>800M) & (V(TD1)>800M) ? {VHIGH} : {VLOW} with V=(V(CLK)>800M) & (V(TD1)>800M) ? {VLOW} : {VHIGH}.

Figure 2a portrays a typical application of the dead-time generator in a simplified half-bridge driver, and **Figure 2b** shows the corresponding IsSpice4 waveforms. You can download both listings from *EDN*'s Web site, www.ednmag.com. Click on "Search Databases" and then enter the Software Center to download the file for Design Idea #2490. (DI #2490)