

DESIGNER'S NOTEBOOK



IDLE2 Instruction on a TMS320C51 When Using a Divide-by-One Clock Option

Contributed by Ramesh Iyer

Design Problem

How does the 'C51 behave when an IDLE2 instruction is executed while the device is in the divide-by-one clock mode? When is the minimum power consumption?

Solution

Execution of the IDLE2 instruction causes a complete shutdown of the core CPU and the on-chip peripherals. The IDLE2 mode is exited by an unmasked interrupt or if an external interrupt is held low for at least five clock cycles.

In the divide-by-one clock mode, an external frequency source can be used to inject the frequency into the CLKIN2 pin, with the pin X1 left unconnected, pin X2 connected to V_{DD} , CLKMD1 pin tied high, and CLKMD2 strapped low.

After a clock is supplied on the CLKIN2 pin, the internal PLL goes through a transitory phase during which it locks in to the supplied clock. Table A-8 of the *TMS320C5x Users Guide* (SPRU056) specifies t_p to be between 256 and 1000 cycles.

The footnote to Table A-8 also specifies that clocks can be stopped only while the device executes IDLE2 when using the external divide-by-one clock option. The footnote to Table A-9 adds that t_p will occur when restarting the clock from IDLE2 in this mode.

The following queries arise:

1. When the 'C51 is in divide-by-one clock mode and the IDLE2 instruction is executed, is the PLL shut down?
2. Occurrence of an external interrupt initiates the transitory phase of 256–1000 cycles and exits IDLE2 mode after five clock cycles. Does this mean that the 'C51 will be executing before the PLL clock has stabilized?
3. What happens if the clock input is not stopped? Will the transitory phase still occur?
4. Which of the following options assures the least power consumption:
 - a) Execute IDLE2, change clock mode to divide-by-two, stop clock
 - b) Execute IDLE2, stop clock
 - c) Execute IDLE2, do not stop clock

When the 'C51 is in IDLE2, the PLL is not shut down.

The transitory phase will NOT occur if the clock input is not stopped. Additionally, the clock should NOT be stopped after an IDLE2 instruction in the divide-by-one clock mode. Tables A-8 and A-9 are incorrect in stating it otherwise.

After execution of an IDLE2 instruction, the clock mode must be changed out of the divide-by-one mode before the clock can be stopped. This will now ensure the least power consumption.

In any case, the clock should never be stopped with the PLL running, failing which, the PLL will start hunting for a clock source to lock on.