

ADSP-2191 PROGRAMMABLE PLL

Created 9/12/2001 - GJO

Overview

The ADSP-2191 includes a *clock multiplier*, which provides more flexibility for multiple clock modes than that of the fixed 2x CLKIN clock mode used by the popular ADSP-218x DSP family. The ADSP-2191's clock multiplier mode enables the chip to run faster than the input clock rate. The PLL can also be brought up at a 1:1 clock ratio (referred to as *Bypass mode*) to reduce power usage. The purpose of this application note is to describe how to properly configure the programmable PLL on the ADSP-2191.

Background

The ADSP-2191 provides a user-programmable 1 to 32 multiplication of the input clock, including some fractional values, to support 128 external to internal (DSP core) clock ratios. The states of the *MSELx*, *BYPASS*, and *DF* pins determine how the PLL will function at reset.

If the *BYPASS* pin is set (=1), the *MSELx* pins are still sampled on the rising edge of /RESET and transition back to programmable flags after the reset is complete; the PLL is put into bypass; and the input clock is directly used to generate the clocks for the core (CCLK) and the peripherals (HCLK). At hard reset the value of the *BYPASS* pin automatically determines the value of the PLLCTL register's *BYPASS* bit; at any other time, software controls this bit.

The *DF* pin (alternately named PF7) controls the input divider. The input divider is disabled when *DF* is cleared (=0). When *DF* is set (=1), CLKIN is divided by two prior to being used to generate the core and the peripheral clocks. At hard reset the value of the *DF* pin automatically determines the value of the PLLCTL register's *DF* bit; at any other time, software controls this bit.

NOTE: DF must be set (=1) when CLKIN is above 100Mhz. This will ensure that HCLK never exceeds 100Mhz.

Controlling the PLL

The ADSP-2191 *MSELx* and *DF* pins are shared and can be used in three different ways: as clock multiplier select pins, general-purpose flag pins (referred to as PF_x or PF [7:0]), or as select pins for external SPI devices. Because of this, depending on how the programmable flags are to be used, certain systems considerations must be made to ensure the *MSELx* and *DF* pins are properly configured and defined when powering-up and hard-resetting the DSP.

When /Reset is asserted (active low), the states of *MSELx*, *DF*, and *BYPASS* are used to initialize the PLL. The core clock and peripheral clock will reflect the states of these pins. A requirement of the PLL multiplier is that the *MSELx* and *DF* must be in a valid configuration ratio for a period not exceeding 15 CLKIN cycles following the assertion of /Reset. Depending on how these pins are implemented in your system (ex: PF's or SPI selects), variables such as capacitive loading and line terminations could negatively affect the *MSELx* and *DF* pins, resulting in an unwanted or invalid multiplication ratio.

Consider a system with a 16 MHz CLKIN and a desired core frequency of 160 MHz. In this case, the desired multiplication ratio would be 10 (PF [7:0] = 0x0A). Given the 15 CLKIN cycle requirement, after the assertion of /Reset, a maximum duration of 937 nsec (15/CLKIN = 15/16M) may pass before the state of PF [7:0] must assume a valid ratio.

NOTE: The PLL is incapable of recovering from invalid multiplication ratios (MSELx/DF = 0x00 or 0xFF) applied for periods exceeding 15 CLKIN cycles.

In some applications, PF [7:0] maybe configured as outputs. Following the assertion of Reset, there is a maximum delay of 15ns before PF [7:0] is tri-stated.

Given below are three system scenarios and suggested system considerations given each scenario.

Scenario 1

In a system, the user doesn't intend to use PF [7:0] for anything other than configuring the multiplication ratio during /Reset. In this case, *MSELx* and *DF* can be tied high or low to achieve the desired multiplication ratio.

Scenario 2

In a system, the user intends to use PF [7:0] for tasks other than *MSELx* and *DF* selection. As an example, connected to an external device such as an SPI slave. In this case, *MSELx* and *DF* can be pulled high or low with a suitable resistor to achieve the desired multiplication ratio. This is ok provided the devices connected to the *MSELx* and *DF* pins do not drive the PF pins during RESET. In some systems this may require external logic to disable the outputs of other devices that would want to over power the resistors.

As shown in Figure 1, simple pull-ups and pull-downs are added to *MSELx* and *DF* pins.

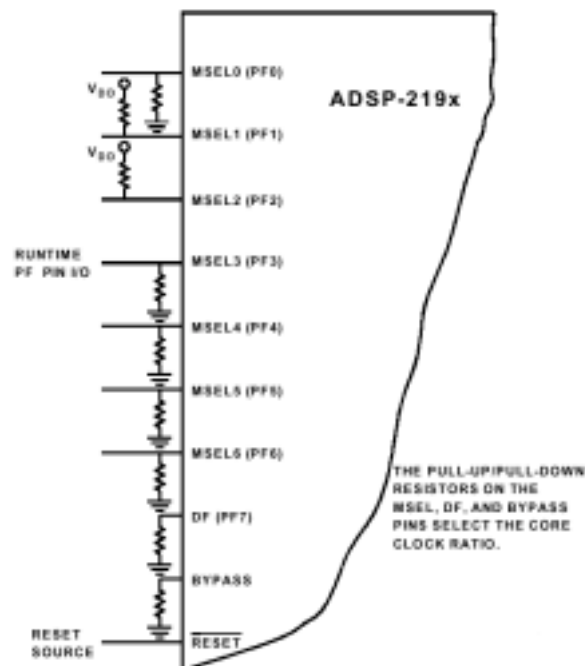


Figure 1. MSELx and DF pin termination.

NOTE: Individual System requirements must ensure the MSELx and DF pins remain invalid no longer than 15 CLKIN cycles after the falling edge of /RESET.

Scenario 3

In a system, the user intends to have PF [7:0] connected to external devices, however the option of having pull-ups/pull-downs on *MSELx* and *DF* as shown in Figure 1 is not available. Some systems must have MSEL driven

during and after reset. In other systems external capacitive loading was so high that pull ups and pull downs are too slow for the requirement for *MSELx* and *DF* pins to be valid 15 CLKIN cycles after the start of RESET asserted low.

In this case, it is suggested that the system incorporate a multiplexing circuit to provide the correct *MSELx* values during and after reset. This will ensure that the *MSELx* timing requirements are satisfied during RESET.

An alternative solution to the above scenario is to put the PLL in *Bypass* mode (tie *BYPASS* high). After /RESET has been de-asserted, software can configure the appropriate *MSELx* bits in the PLLCTL register and then clear the *BYPASS* bit. The PLL will then assume the new PLLCTL settings. As shown in Figure 2, a simple pull-up is added to *BYPASS*.

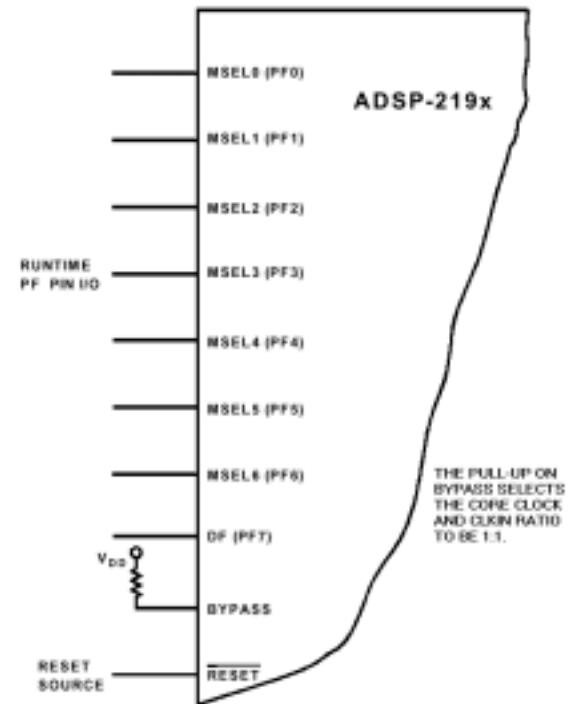


Figure 2. BYPASS pin termination.

NOTE: In some systems using BYPASS mode, depending on the chosen boot mode, booting a large program may be very slow. In Host boot mode, the BYPASS method should be considered an easy workaround for using the PF pins for application specific functions without requiring any MSELx considerations during reset.

Software Configuration

The code listed below demonstrates the instructions needed to change the clock mode of the 2191 from *Bypass* to *Multiply*. The code was verified with the BYPASS pin active (tied high).

NOTE: When in bypass mode: cclk=clkin if DIV2=0 (bit7 of the PLL CTL reg) and cclk=clkin/2 if DIV2=1.

```
/******  
/* PLL Control Register */  
  
iopg = 0x00;      /* Initialize IOPG to Clock and System Control */  
  
ax1 = 0x0552;    /* Stop the PLL - In Bypass */  
io(0x200) = ax1; /* Write the PLLCTL register */  
nop;  
  
ax1 = 0x0352;    /* Reprogram to 1x - In Bypass */  
io(0x200) = ax1; /* Write the PLLCTL register */  
nop;  
  
ax1 = 0x0350;    /* Start the PLL - In Bypass */  
io(0x200) = ax1; /* Write the PLLCTL register */  
nop;  
  
ax1 = 0x0250;    /* Come out of Bypass- Takes ~500 clkin cycles */  
io(0x200) = ax1; /* Write the PLLCTL register */  
  
cntr = 1100;     /* Wait for a duration of time */  
do wt until ce;  
wt:  nop;  
/******
```