Clocks Setup for MFC-VXI Module Cogging Application

# Available Clocks

The MFC-VXI module developed by the AD/RF Group is being applied to the Booster Cogging application. The MFC module has a number of logic clocks provided for the DSP and programmable logic, and also clocks for controlling the ADC sampling and DAC output update rate.

Figure I.1 is a block diagram that indicates the available clocks and there distribution on the module.

Clock oscillator, U6, provides a 31.24MHz clock. This clock is used as the logic clock for the VXI/VME interface and other logic in the MAX II programmable logic. The same clock is multiply up to higher frequencies by the Sharc DSP PLL for use as its core processing clock, at 328.02 MHz, and the clock used by the DSP’s SDRAM interface and its Asynchronous Memory Interface (AMI). The 131.28 MHz clock used for the memory interface is the clock considered when defining the duration of Wait states and Hold cycles define for interfacing to the various memory resources (Flash, FPGA, MAX II).

The FPGA also receives a copy of the 31.24 MHz clock, but at this point in time it is not the choice for the logic clock. Rather, a clock from the AD9510 clock distribution chip, derived from the 65 MHz clock oscillator, is preferred. The preference is based on the fact that the ADC sampling clocks and the DAC update clocks are also derived from the 65 MHz clock, through the clock distribution chip. A logic clock in the FPGA that is coherent with the ADC and DAC clocks would be convenient to keep the interfaces between these synchronous. The interface between the FPGA and the DSP and MAX II VME interface is solely across the address/data bus. Here, the data transfers with the FPGA do not need to be so fast that the extra wait and hold states needed to synchronize the signals will be impact performance.

Figure I.1 Indicates that the ADC and DAC clocks are the 65 MHz clock divided by 16 (4.0625 MHz) and that the DAQ\_Clock in the FPGA is divided down to this value also. The divide by 16 here is arbitrary. The point is that DAQ\_Clock in the FPGA will be adjusted to be synchrous with the ADC and DAC clocks. If there is more than a single rate for the ADC / DAC clocks, the FPGA has enough PLL outputs to replicate more.

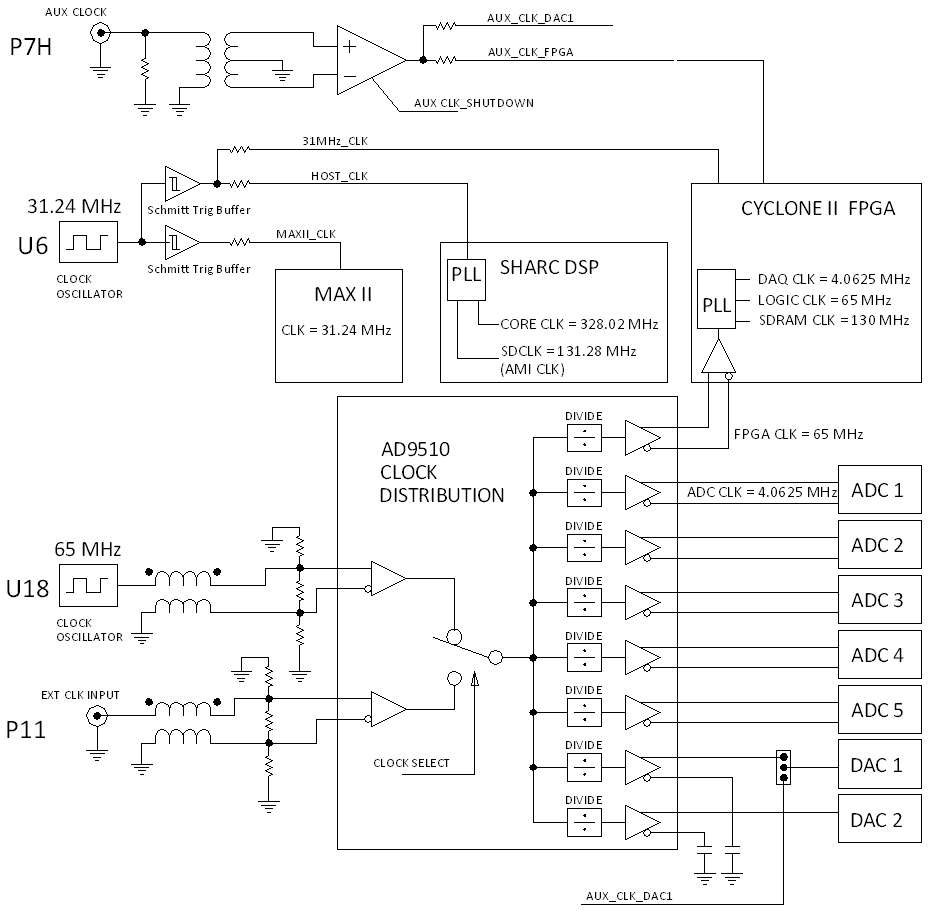


Figure I.1 Clock Distribution Block Diagram

# The Clock Distribution Chip Setup

The AD9510 clock distribution chip is configured by the DSP processor after power is applied to the module. The settings for the clock distribution chip configuration registers are defined in the DSP code and given in Table I.1

The 32 bit words loaded into the serial port of the AD9510 Clock Distribution chip consist of 16 bit Instruction Word (MSB), followed by two 8 bit Data Bytes. The most significant data byte is the setting for the register addressed in the Instruction Word (N). The last data byte is the setting for the register previous to that addressed in the Instruction Word (N-1).

The layout of the Instruction Word is shown below.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| I15 | I14 | I13 | I12 | I11 | I10 | I9 | I8 | I7 | I6 | I5 | I4 | I3 | I2 | I1 | I0 |
| R/W# | W1  (=0) | W0  (=1) | A12  (=0) | A11  (=0) | A10  (=0) | A9  (=0) | A8  (=0) | A7  (=0) | A6 | A5 | A4 | A3 | A2 | A1 | A0 |

When setting up the clock distribution chip we are writing the registers so the R/W# bit would be 0. Considering the other default values, we have the Instruction Word with the following form.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| I15 | I14 | I13 | I12 | I11 | I10 | I9 | I8 | I7 | I6 | I5 | I4 | I3 | I2 | I1 | I0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | A6 | A5 | A4 | A3 | A2 | A1 | A0 |

Here is an example of a 32 bit setup word

Example: 0 x 2 0 4 D 8 0 7 7 (hexidecimal)

Write operation I15 = 0

2 data bytes follow so [I14, I13] = [0, 1]

A[12...8] = 0 (the default value for these reserved bits)

So we have 0010 0000 (0x20) for the first byte (MSB)

The second byte I[7...0] is then the register (N) address, 0x4D.

The 2 data bytes in the example above are 0x80 to be written to register 0x4D (N)

and 0x77 to be written to register 0x4C (N-1).

See the AD9510 datasheet for definition of the data bytes and register bits.

Table 1 Sample setup for the AD9510 Clock Distribution Chip

|  |  |
| --- | --- |
| 0x2000 9000 | 1. Makes SDIO pin bidirectional and SDO inactive |
| 0x203D 0808 | (3D) Output 0: PECL logic, on, .810 V output level ADC2 |
|  | (3C) Output 1: PECL logic, on, .810 V output level ADC1 |
| 0x203F 0808 | (3F) Output 3: PECL logic, on, .810 V output level ADC3 |
|  | (3E) Output 2: PECL logic, on, .810 V output level ADC4 |
| 0x2041 0A0A | (41) Output 5: CMOS logic, on,3.5 ma output level, 100 ohms Termination, DAC2 |
|  | (40) Output 4: CMOS logic, on,3.5 ma output level, 100 ohms Termination, DAC1 |
| 0x2043 0202 | (43) Output 7: LVDS logic, on,3.5 ma output level, 100 ohms Termination, FPGA |
|  | (42) Output 6: LVDS logic, on,3.5 ma output level, 100 ohms Termination, ADC5 |
| 0x2045 0100 | (45) Input Clock: 01 - CLK1, 65 MHz crystal || 00 - CLK2, LO External Clock |
|  | (44) Address 0x44 not used |
| 0x2049 0011 | (49) Output 0: phase=0, ADC2 clock |
|  | (48) Output 0: Divide ratio 4, ADC2 clock |
| 0x204B 0011 | (4B) Output 1: phase=0, ADC1 clock |
|  | (4A) Output 1: Divide ratio 4, ADC1 clock |
| 0x204D 0011 | (4D) Output 2: phase=0, ADC4 clock |
|  | (4C) Output 2: Divide ratio 4, ADC4 clock |
| 0x204F 0011 | (4F) Output 3: phase=0, ADC3 clock |
|  | (4E) Output 3: Divide ratio 4, ADC3 clock |
| 0x2051 8000 | (51) Output 4: phase=0, DAC1 clock |
|  | (50) Output 4: Divide ratio 2, Divider Bypassed DAC1 clock |
| 0x2053 8000 | (53) Output 5: phase=0, DAC2 clock |
|  | (52) Output 5: Divide ratio 2, Divider Bypassed DAC2 clock |
| 0x2055 0011 | (55) Output 6: phase=0, ADC5 clock |
|  | (54) Output 6: Divide ratio 4, ADC5 clock |
| 0x2057 8000 | (57) Output 7: phase=0, FPGA clock |
|  | (56) Output 7: Divide ratio 2, Divider Bypassed FPGA clock |
| 0x2058 2580 | (58) Function pin: SYNCB |
|  | (57) Output 7: phase=0, FPGA clock |
| 0x205A 0100 | (5A) Transfer buffer data to control registers |
|  | (59) Address 0x59 not used |