APPLICATION NOTE

RGB Palette DAC PLL Programming

by David R. Warfield

Introduction

Every member of the RGB family of Palette DACs contains a programmable clock synthesizer for generating the clock used by the DAC circuits and other components of a graphics adapter for pixel and CRT timings. This clock is known as the DOT clock or PIXEL clock.

The clock generator is a Programmable Phase-locked Loop (PLL) that operates off a fixed external reference clock. With appropriate programming, the PLL uses this fixed reference to produce the desired dot clock with a frequency ranging from approximately 16 MHz to 256 MHz.

Some Palette DACs contain a second PLL for generating a general purpose “SYSCLK” (also known as MCLK). The supported frequency range for the SYSCLK PLL output is typically 16 MHz to 100 MHz; otherwise the programming of the two PLLs is identical.

There are three different methods of programming the PLLs:

1. “Direct”.
2. “Old style M/N” (M over N, also called “pseudo” M/N, or “restricted” M/N).
3. “New style M/N” (also called “standard” or “full” M/N).

The first members of the RGB Palette DAC family use an “old style” PLL. Newer members contain an enhanced PLL that supports the “new” M/N programming style. Table 1 shows the programming types available by product.

The product data sheets contain specifications for the external reference clock, output frequency ranges, programming type, programming register usage, and programming formulas. This application note explores the operation of the PLLs in more detail, and shows the derivation of these formulas. This information can be used:

- For a better understanding of the programming formulas and the trade-offs that can be applied.
- To operate the PLL such that jitter is minimized.

Table 1. Programming Types by Product

Product	PLL Generation	Programming Types
RGB514, RGB524, RGB525, RGB528A, RGB561	Old Style	1. Direct 2. Old Style M/N
RGB526/RGB526DB, RGB624/RGB624DB	New Style with backward compatibility to RGB51x, RGB52x	1. Direct 2. Old Style M/N 3. New Style M/N
RGB640	New Style	1. New Style M/N

PLL Principles of Operation

A simplified diagram of a programmable PLL is shown in Figure 1. The operation is standard:

1. A phase comparator compares two frequencies and generates a difference signal. One of the frequencies is a fixed reference (f_{INTREF}).
2. The phase comparator drives a charge pump which develops a difference signal on the capacitor of a filter. The filtered difference signal drives a voltage-controlled oscillator (VCO).
3. The output of the VCO is divided down and fed back to the phase comparator for comparison against the reference.
4. If the value of the VCO divider circuit is “M”, then when

   $$f_{VCO} = M \times f_{INTREF}$$

   the divided input to the phase comparator will be the same as the reference, which will make the difference signal = zero, which will cause the VCO to stay where it is. If the VCO tends to drift for any reason a difference signal will be generated which will drive the VCO back to where it should be ($M \times f_{INTREF}$). Thus, the output frequency is “locked” to the reference frequency.
5. The incoming reference clock (REFCLK) is prescaled by the value N to provide the internal reference f_{INTREF}. A postscaler value of P is used to divide down the VCO output frequency to the final output frequency of the CLOCK OUT signal.
6. The relation of the final output frequency to the incoming reference frequency is:

   $$f_{CLOCK\ OUT} = f_{REFCLK} \times (M / (N \times P))$$

The values of M, N and P are provided by registers which can be altered from a microprocessor bus. The PLL is programmed to a desired frequency by writing appropriate values to these registers.

Operational Limits

For correct operation of the PLL four frequencies must be checked:

1. The incoming reference clock frequency f_{REFCLK} has a minimum and maximum frequency.

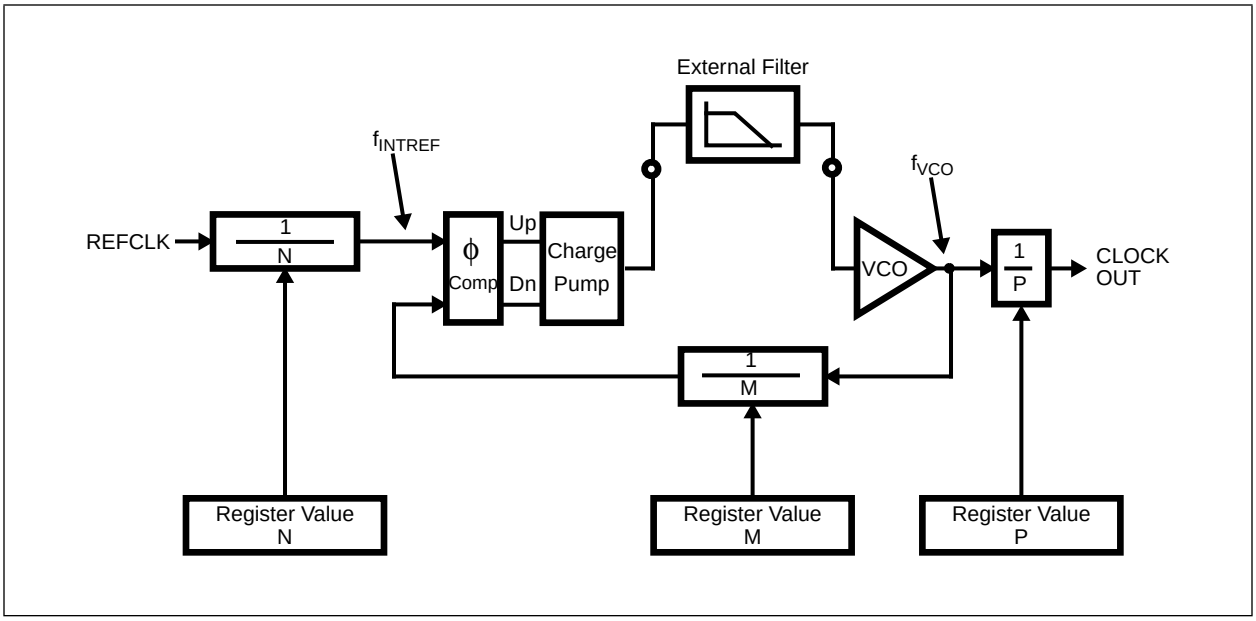


Figure 1. Generic PLL

2. The internal reference to the phase comparator f_{INTREF} has a minimum frequency. In general an f_{INTREF} of less than 1 MHz will cause excessive jitter.
3. The VCO has a minimum and maximum frequency. In general the VCO minimum frequency is 65 MHz, and the maximum frequency is usually the same as the maximum CLOCKOUT frequency.
4. CLOCKOUT has a maximum frequency. For the pixel PLL the maximum frequency is the maximum clock speed of the product (170 MHz, 220 MHz, …), and for the SYSCLK PLL the maximum frequency is typically 100 MHz.

The Direct Programming method used with the “old style” PLLs require the incoming REFCLK to be constrained to certain frequencies. If this condition is met then the only condition that has to be checked is that CLOCKOUT does not exceed the specified maximum frequency. The internal frequencies f_{INTREF} and f_{VCO} are guaranteed correct.

The “pseudo” M/N programming method used with the “old style” PLL and the “true” M/N programming method used with the “new style” PLL eases the REFCLK constraints, but illegal f_{INTREF} and f_{VCO} frequencies can be generated, and so must be checked.

The numbers given above and elsewhere in this application note are for illustration only; the product datasheets listed in the Table 2 must be used for the actual PLL specifications:

Table 2. Product Data Sheets

Product	Data Sheet Document Number
RGB514	SC22-9860
RGB524	SC22-9861
RGB525	SC22-9863
RGB526/ RGB526DB	SC22-9862
RGB528A	SC22-9864
RGB561	SC22-9866
RGB624/ RGB624DB	SC22-9867
RGB640	SC22-9865

Jitter Reduction

The programmable PLL is excellent for “dialing in” any of a wide variety of pixel clock frequencies. However, PLLs tend to have more jitter than fixed crystal oscillators, and unfortunately even a small amount of jitter can be seen easily on displays.

It is difficult to say just what is an acceptable amount of jitter, as some monitors are more susceptible than others, and some frequencies give more trouble than others.

The primary means of jitter control is to provide a good electrical environment to the PLL (clean VDD, no noise coupled into the loop filter). In general this is sufficient, and the PLL can be programmed without regard to jitter. However, jitter can be reduced further still with appropriate programming values, if desired.

A key factor in PLL jitter is the phase comparator (Figure 1). Essentially this circuit “bumps” the VCO frequency up or down in tiny amounts, based on the time difference between the leading edges of the two signals it is comparing (f_{INTREF} and the divided VCO output). During the time between edges, the VCO can drift away from the lock frequency. So if the edges of the two signals arrive closer together (they are higher in frequency) there will be less drift, or jitter, of the VCO frequency around the lock frequency.

In other words the higher the f_{INTREF} the less jitter.

It is for this reason that when using M/N programming, f_{INTREF} must be checked for operation no less than 1 MHz. Anything less may produce unacceptable jitter.

In principle there is no upper limit on f_{INTREF}, but because the VCO frequency is $M \times f_{INTREF}$, an upper limit is quickly reached based on the maximum VCO frequency.

In generating programming values for the PLLs it is possible to come up with more than one set of values that produce the same frequency. If this occurs, f_{INTREF} for each combination should be checked, and the combination that produces the highest f_{INTREF} (and legal VCO frequency) should be used.

Postscaler

The scaler following the VCO, controlled by the value "P", is generally used to produce CLOCKOUT frequency values less than the minimum VCO frequency. E.g., to produce a 40 MHz clock the PLL would be programmed to have a VCO frequency of 80 MHz, and the P value would be set to divide the VCO output frequency by 2.

When the VCO frequency is divided down any jitter is also divided down, therefore for jitter reduction it is desirable to have the VCO run higher than the desired CLOCKOUT frequency. This tends to go hand-in-hand with the policy of having a high f_{INTREF} to minimize jitter. However the technique is somewhat limited because at higher frequencies the VCO will exceed its specified maximum frequency.

SYSCLK PLL Jitter

In general the Pixel PLL is used to clock directly or indirectly all signals going to the display monitor, such as the horizontal and vertical sync signals, as well as the video data (pixel) signals. Excessive jitter on these signals will be readily evident to the human eye.

For the SYSCLK PLL jitter control may not be so important, and must be evaluated on an application by application basis. Where jitter must be controlled, the techniques described above apply to both the Pixel PLL and the SYSCLK PLL.

"OLD Style" PLL

A simplified diagram of the Old Style PLL is shown in Figure 2. This PLL is optimized for the Direct Programming method, and differs from the generic PLL of Figure 1 in the following details:

1. In the product data sheets the internal reference frequency f_{INTREF} is identified as VRF (Video Reference Frequency).
2. The "N" value is called REF DIV COUNT (Reference Divider Count). A register provides 5 bits for this value, yielding a maximum divide value of 31. The values 0 and 1 are illegal.
3. The "M" value is called VCO DIV COUNT (VCO Divider Count). 6 bits are provided, and VCO DIV COUNT can range from 0 through 63. The value 65 is added to VCO DIV COUNT, yielding a VCO divider value ranging from 65 through 128.
4. The same register holding the VCO DIV COUNT value also holds a 2 bit value called DF (Desired Frequency). These two bits are decoded to control the following functions:

   a. Following the 1/(2 to 31) prescaler, there is an additional divide-by-two circuit. The DF bits select whether VRF, the input to the phase comparator, uses the prescaler directly or if it has the additional divide-by-two. If the divide circuit is chosen, then VRF = 1/(4 to 62, in steps of 2) × REFCLK.

   b. The postscaler on the output of the VCO is controlled by the DF bits. The DF bits effectively act as the "P" value, and select whether CLOCKOUT is the direct output of the VCO, the VCO divided by 2, or the VCO divided by 4.

   c. The charge pump operation must be adjusted depending on the frequency of operation. The DF bits are used for this.

      Further discussion of this topic involves detailed explanation of the charge pump, the damping factor of the loop filter, and operation of the VCO, which is beyond the purview of this application note. For operation within the specified frequency range of the product (170/220/250 MHz) the effect of DF on the charge pump can be ignored.

Table 3 shows the effect of the DF bits on the divide-by-two circuit and the postscaler:

Table 3. DF Bit Action

DF	Prescaler Input to VRF	Output of VCO
00	÷ 2	÷ 4
01	÷ 2	÷ 2
10	÷ 2	÷ 1
11	÷ 1	÷ 1

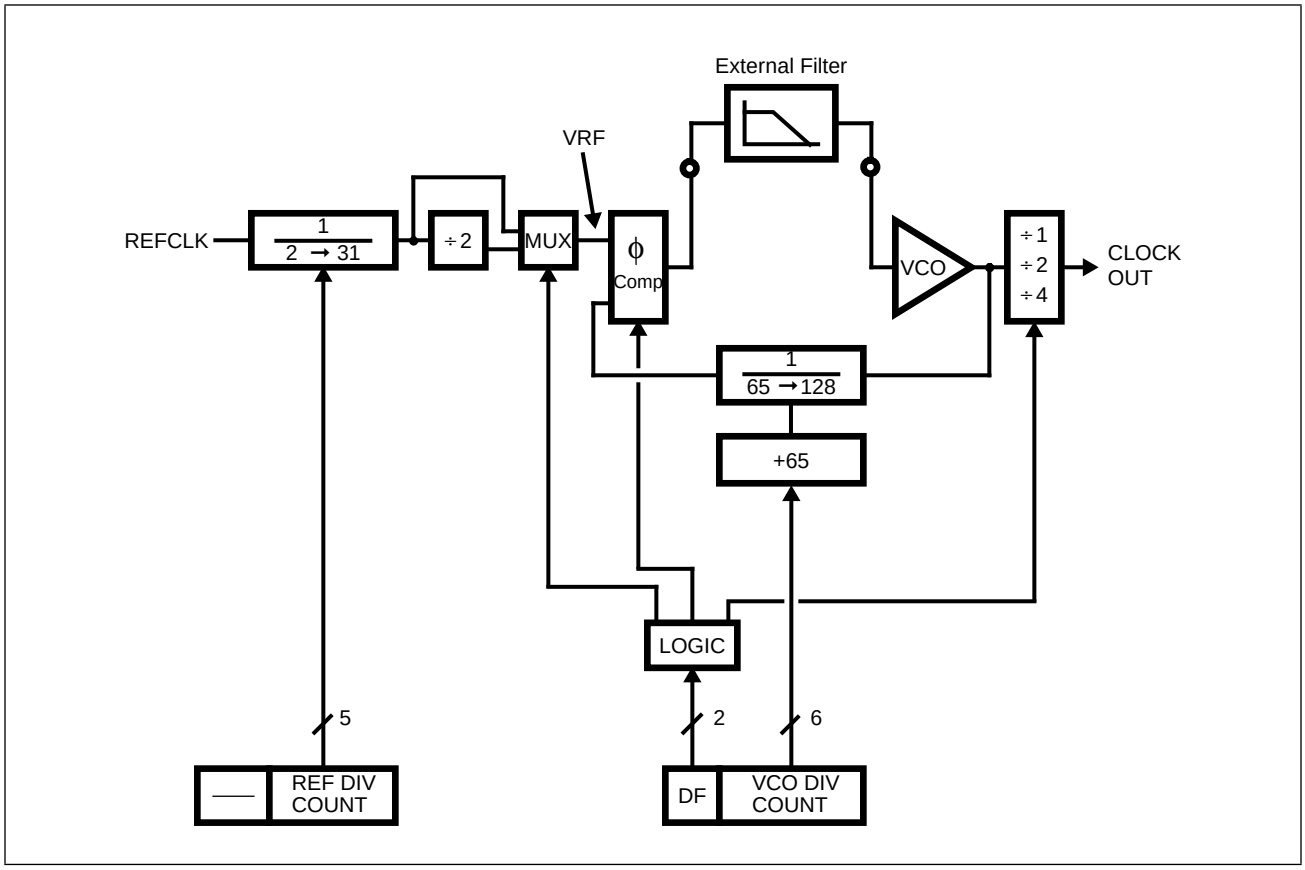


Figure 2. Old Style PLL

Using this information, the values of the internal reference VRF and CLOCK OUT can be calculated, as shown in Table 4:

Table 4. Old Style PLL Formulas for VRF and CLOCK OUT

DF	VRF	$f_{CLOCK\ OUT} = f(VRF)$	$f_{CLOCK\ OUT} = f(f_{REFCLK})$
00	$\frac{f_{REFCLK}}{(\text{REF DIV COUNT}) \times 2}$	$\frac{VRF \times (\text{VCO DIV COUNT} + 65)}{4}$	$\frac{f_{REFCLK} \times (\text{VCO DIV COUNT} + 65)}{(\text{REF DIV COUNT}) \times 8}$
01	$\frac{f_{REFCLK}}{(\text{REF DIV COUNT}) \times 2}$	$\frac{VRF \times (\text{VCO DIV COUNT} + 65)}{2}$	$\frac{f_{REFCLK} \times (\text{VCO DIV COUNT} + 65)}{(\text{REF DIV COUNT}) \times 4}$
10	$\frac{f_{REFCLK}}{(\text{REF DIV COUNT}) \times 2}$	$VRF \times (\text{VCO DIV COUNT} + 65)$	$\frac{f_{REFCLK} \times (\text{VCO DIV COUNT} + 65)}{(\text{REF DIV COUNT}) \times 2}$
11	$\frac{f_{REFCLK}}{\text{REF DIV COUNT}}$	$VRF \times (\text{VCO DIV COUNT} + 65)$	$\frac{f_{REFCLK} \times (\text{VCO DIV COUNT} + 65)}{\text{REF DIV COUNT}}$

Table 5. Direct Programming Reference Divider Values

REFCLK (MHz)	Fixed PLL Reference Divider Register Value
4	0x02
6	0x03
8	0x04
10	0x05
12	0x06
14	0x07
16	0x08
18	0x09
20	0x0a
22	0x0b
24	0x0c
26	0x0d
28	0x0e
30	0x0f
32	0x10
34	0x11
36	0x12
38	0x13
40	0x14
42	0x15
44	0x16
46	0x17
48	0x18
50	0x19
52	0x1a
54	0x1b
56	0x1c
58	0x1d
60	0x1e
62	0x1f

Direct Programming

With “direct programming” the internal comparator reference VRF is held at a fixed frequency. This allows a single value to be used for prescaler, and this value can be obtained from a simple table lookup against the REFCLK frequency. Various VCO DIV COUNT values are then used to produce the desired frequencies.

In practice the PLL uses four frequency ranges, set by the DF bits. The three lowest frequency ranges work with an internal VRF value of 1 MHz. The prescaler value is chosen to produce a 2 MHz frequency just ahead of the divide-by-two circuit. Then, as shown in Table 3 on page 4, the divide-by-two circuit is selected to set VRF at 1 MHz.

The lowest frequency range uses the divide-by-four on the VCO output, which allows a granularity or “step size” of 0.25 MHz (1/4 the VRF frequency). The next frequency range uses divide-by-two on the VCO output for a step size of 0.5 MHz. The third frequency range uses divide-by-one (no divide) for a step size of 1 MHz.

The fourth frequency range bumps the internal VRF up to 2 MHz. The DF bits cause the divide-by-two on the input prescaler to not be used, so the 2 MHz output of the prescaler is used directly as VRF. No divide is used on the VCO output, and the step size is the VRF frequency of 2 MHz.

With the prescaler range of 2 to 31 and the requirement to produce a 2 MHz reference ahead of the divide-by-two circuit, the incoming REFCLK frequency is required to be from 4 to 62 MHz on 2 MHz boundaries. This produces Table 5. The table can be expressed by the equation:

$$f_{REFCLK}/(\text{REF DIV COUNT}) = 2$$

Knowing this, using the column in Table 4 on page 5 labeled “$f_{CLOCK\ OUT} = f(f_{REFCLK})$” and solving for VCO DIV COUNT as a function of CLOCK OUT frequency produces Table 6:

Table 6. PLL Frequency Direct Programming

DF	VCO Divide Count	Frequency Range	Step (MHz)
00	$(4 \times f_{CLOCK\ OUT}) - 65$	16.25 - 32 MHz	0.25
01	$(2 \times f_{CLOCK\ OUT}) - 65$	32-5 - 64 MHz	0.5
10	$f_{CLOCK\ OUT} - 65$	65 - 128 MHz	1.0
11	$(f_{CLOCK\ OUT}/2) - 65$	130 - 256 MHz	2.0

The advantages of direct programming are:

1. Ease of programming.
2. Simple formulas and table lookup.
3. Easy to understand.
4. With straightforward step sizes of 1/4, 1/2, 1 and 2 MHz it is easy to calculate how close the actual frequency is to the desired frequency.
5. On a percentage basis the step size of 1 MHz for 100 MHz frequency gives the same “error” (actual versus desired) as 2 MHz does for 200 MHz.
6. To work with larger ranges of VRF may require more programming of the charge pump current, loop filter value, or VCO gain. These would require extra hardware bits. The scheme used on the RGB series requires only 5 bits for the prescaler and 8 bits for each frequency specified by the DF and VCO DIV COUNT bits.

Pseudo M/N

The direct programming scheme has three disadvantages:

1. It may be better to use a higher VRF than 1 MHz to further minimize PLL jitter in a high noise environment.
2. It may be desired to use a REFCLK frequency that does not meet the specification of 4 MHz to 62 MHz on a 2 MHz boundary.
3. For special bids in which the desired pixel clock is greater than 256 MHz, direct programming cannot produce the required values.

As discussed previously, “M over N” is so-called from the concept that

$$\text{PLL Output Frequency} = \text{Input Frequency} \times (M/N)$$

in which VCO DIV COUNT + 65 provides M, REF DIV COUNT provides N, and the equation is modified by the actions of the DF bits controlling the VRF divide-by-two and the VCO postscaler.

Table 4 on page 5 contains the actual equations. In general a heuristic algorithm is used to calculate the programming values. The direct programming values can be used as a starting point, as described in below.

“Old Style” Programming Examples

For the purposes of this example we’ll suppose the desired CLOCK OUT frequency is 65 MHz, and the incoming REFCLK is 50 MHz.

Direct Programming Method

The REF DIV COUNT is a simple lookup. For a 50 MHz REFCLK we need a value of 25 (Table 5). This will produce a frequency of 2 MHz just ahead of the divide-by-two, and will be used for any frequency we generate.

In Table 6 we see that 65 MHz is just inside the third frequency range, indicating that the DF bits should be `10`.

Also from that table, the VCO DIV COUNT = $f_{CLOCKOUT}$ - 65. With a desired $f_{CLOCKOUT}$ of 65 MHz, this produces a value of 0 for VCO DIV COUNT.

Pseudo M/N Programming Method

From the given description of the PLL operation, we can see that with direct programming the DF bits will cause the divide-by-two circuit to be selected, VRF will be 1 MHz, and the VCO will be operating at the bottom of its range at 65 MHz.

Now suppose we want to raise VRF, perhaps to further reduce jitter. As a first step let’s raise VRF to 2 MHz. If we leave VCO DIV COUNT at 0, the VCO will still operate at VRF × 65, or 130 MHz. We want to divide this by 2 to get back to 65 MHz, and we can do this by setting DF = `01`.

With DF = `01`, the divide-by-two circuit will still be active. To get VRF = 2 MHz, we need 4 MHz out of the reference divider. With the 50 MHz REFCLK we can’t do this (we need REF DIV COUNT = 50/4 = 12.5, which can’t be programmed). But if we change REFCLK to 48 MHz, than a REF DIV VALUE of 12 will produce 48/12 = 4 MHz.

If changing REFCLK to 48 MHz is acceptable we could stop here. But what if 50 MHz is a definite requirement? In this case, we can fiddle with programming values and see how close we come.

The ideal REF DIV COUNT is 12.5; the practical values are either 12 or 13. If we use 12, VRF will be (50/12)/2 = 2.083 MHz. The VCO frequency will be 2.083 × 65 = 135.4 MHz, and the CLOCK OUT will be 135.4/2 = 67.7 MHz.

To get closer to 65 MHz we'd have to have a VCO DIV COUNT value of less than 0, which is not possible. So we're as close as we can get with REF DIV COUNT = 12. The difference is about 4%.

Setting REF DIV COUNT to 13 produces VRF = (50/13)/2 = 1.923 MHz. Now the output frequency will be lower than desired. With the VCO DIV COUNT of 0, the VCO output will be 1.923 × 65 = 125 MHz, and CLOCK OUT = 125/2 = 62.5 MHz.

The closest we come is when VCO DIV COUNT = 3. This produces VCO output = 1.923 × (3+65=68) = 130.76 MHz, and CLOCK OUT = 130.76/2 = 65.38 MHz. The difference is now only 0.6%.

Both values that we've looked at are higher than the desired value. VCO DIV COUNT = 2 produces VCO output = 1.923 × (2+65=67) = 128.84 MHz, and CLOCK OUT = 128.84/2 = 64.42 MHz which is lower than 65 MHz, with the difference = 1.6%.

VRF is now approximately twice the value it has with direct programming values, and the VCO is running twice as fast, both of which are good for jitter reduction.

"Old Style" Example Summary

1. For the given REFCLK of 50 MHz the direct programming method can produce the desired 65 MHz exactly. Internally, VRF will be 1 MHz, and the VCO frequency will be 65 MHz.
2. If the REFCLK can be changed to 48 MHz, M/N programming will also produce 65 MHz exactly. Internally VRF will be 2 MHz, and the CLOCK OUT will be 130 MHz divided down by 2. Both of these factors are better for jitter.
3. If it is required to have a REFCLK of 50 MHz, M/N programming can be used to come close to the 65 MHz CLOCK OUT while still maintaining a nearly 2 × VRF over the direct programming value of 1 MHz.

The highest VRF (2.083 MHz) produces a CLOCK OUT 4% higher than the desired 65 MHz. The slightly lesser VRF (1.923 MHz) produces a CLOCK OUT either 1.6% less or 0.6% more than 65 MHz.

The choice of programming values depends on the display monitor sensitivity to jitter and tolerance for deviation from the "correct" CLOCK OUT frequency.

"New Style" PLL

A simplified diagram of the New Style PLL is shown in Figure 3. Compared to the generic PLL of Figure 1, the actual PLL has the following characteristics:

1. The N value is represented with 6 bits. A '1' is added to this value to prevent an illegal divide value of '0'. This yields divide values of the range 1 through 64.

$$f_{INTREF} = \frac{f_{REFCLK}}{N+1}$$

2. 7 bits are used for the M value. A '1' is added to prevent an illegal divide value of '0'. However, only M values of the range 2 through 127 should be used. This yields a divide range of 3 through 128.

$$f_{VCO} = f_{INTREF} \times (M+1) = f_{REFCLK} \times \frac{M+1}{N+1}$$

3. The P value is represented with 3 bits, and decoded to provide a divide value of 1 when P is 0, and 2 × P when P = 1, 2, 3 or 4. Other values of P are illegal.

$$f_{CLOCKOUT} = f_{REFCLK} \times \frac{M+1}{N+1};\ P = 0$$

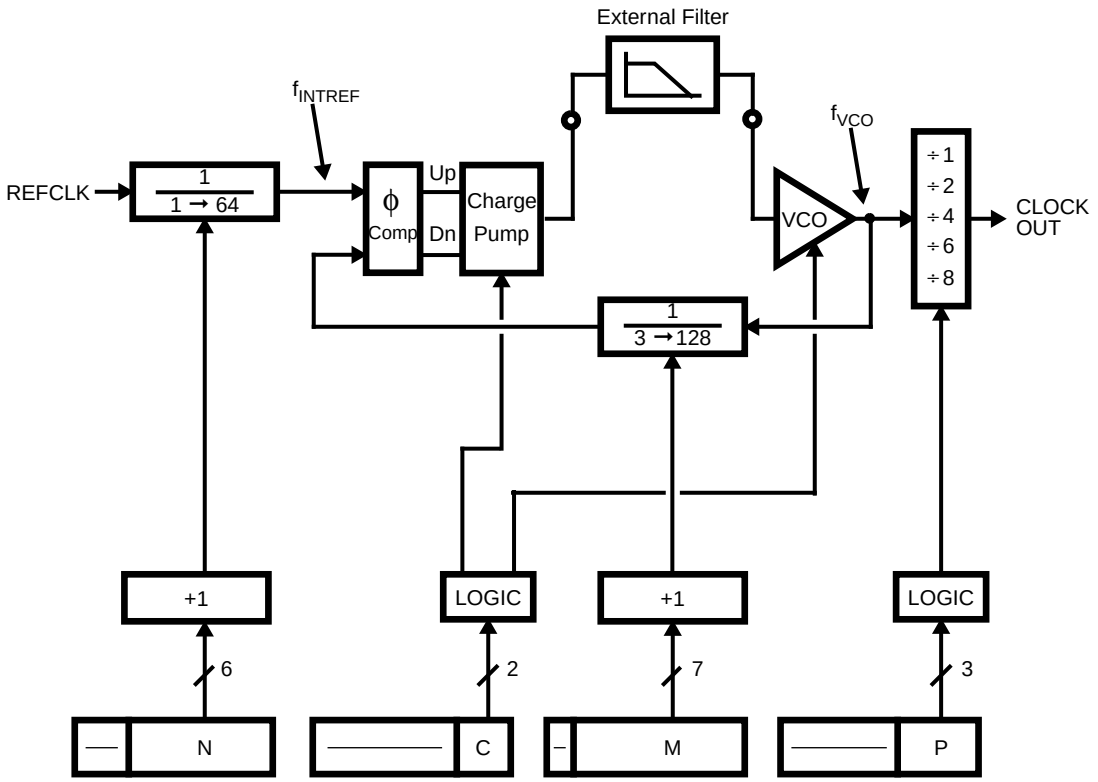
$$f_{CLOCKOUT} = f_{REFCLK} \times \frac{M+1}{(N+1) \times 2P};\ P = 1, 2, 3, 4$$

4. A 2-bit value C is added to control the charge pump bias current and VCO gain.

$$\text{set } C = 1; \text{ when } 65\ \text{MHz} \le f_{VCO} \le 128\ \text{MHz}$$

$$\text{set } C = 2; \text{ when } f_{VCO} > 128\ \text{MHz}$$

Other values of C are reserved.



Input Values	$1\ \text{MHz} \le f_{REFCLK} \le 100\ \text{MHz}$ $2 \le M \le 127$ $0 \le N \le 63$ $0 \le P \le 4$
Output Frequency	$65\text{MHz} \le \left(f_{CLOCKOUT} = f_{REFCLK} \times \frac{M+1}{N+1};\ P = 0\right) \le f_{CLOCKOUT\ MAX}$ $32.5, 16.25, 10.83, 8.125\ \text{MHz} \le \left(f_{CLOCKOUT} = f_{REFCLK} \times \frac{M+1}{(N+1)\times 2P};\ P = 1, 2, 3, 4\right) \le f_{CLOCKOUT\ MAX}$
Requirements for Correct Operation	$1\ \text{MHz} \le \left(f_{INTREF} = \frac{f_{REFCLK}}{N+1}\right)$ $65\ \text{MHz} \le \left(f_{VCO} = f_{REFCLK} \times \frac{M+1}{N+1} = f_{INTREF} \times (M+1)\right) \le f_{CLOCKOUT\ MAX}$ For 2 PLLs, $f_{higher} \neq (n \times f_{lower}) \pm 3\ \text{MHz};\ n = 1,2,3\ldots$
Additional input(s) to set operating point	set $C = 1$; when $65\ \text{MHz} \le f_{VCO} \le 128\ \text{MHz}$ set $C = 2$; when $f_{VCO} > 128\ \text{MHz}$

Figure 3. New Style PLL

In comparison to the old style PLL, the new style PLL:

1. Has a greater flexibility in generating various frequencies.
2. Can run with higher internal reference frequencies (f_{INTREF}).
3. Is not as simple to program.

The new style PLL has a larger range of values that can be programmed for M, N and P. In particular, the minimum value of 65 for the M value is eliminated. This lets the internal reference f_{INTREF} run at a higher frequency without causing the VCO frequency to exceed its maximum specification.

The new style PLL also eliminates the restrictive connection between the N and P values (the divide-by-2 on the pre-scaler that is controlled by the DF bits on the old style PLL.)

With the old style PLL it was known by the setting of the DF bits what the range of frequency operation would be, and internally the operating point of the charge pump and VCO was set automatically. With the new style PLL this is no longer true; so an additional programming value, “C” is required to control these analog circuits.

“New Style” Programming Examples

Example 1

The example used for the old style PLL had a reference clock input of 50 MHz, and a desired output frequency of 65 MHz. A 1 MHz f_{INTREF} is easily generated, and multiplying by the minimum M value of 65 yielded a 65 MHz VCO frequency, which divided by 1 was also the desired output frequency.

However, it was difficult to raise f_{INTREF} to minimize jitter. The best that could be done was to divide the incoming 50 MHz reference by 13 to produce f_{INTREF} = 1.923 MHz, multiply that by 68 to have a VCO frequency of 130.76 MHz, and divide that by 2 to yield 65.38 MHz.

With the new style PLL, the minimum M value is 3. This allows us to boost f_{INTREF} even higher. Dividing 50 MHz by 5 produces f_{INTREF} = 10 MHz, multiplying that by 13 gives a VCO frequency of 130 MHz, and dividing that by 2 gives the desired output frequency of 65 MHz.

Using the equations from Figure 3:

Set N = 4. This produces a prescaler value of 4 + 1 = 5.

Set M = 12. This produces a VCO multiplier value of 12 + 1 = 13.

Set P = 1. This produces a postscaler value of $1 \times 2 = 2$.

Set C = 2, for a VCO frequency greater than 128 MHz.

Example 2

A common graphics adapter frequency is 14.31818 MHz. Suppose this is used as the incoming REFCLK, and it is desired to have an output frequency of 220 MHz.

With a prescaler value of 2 we can have an f_{INTREF} = 7.159 MHz. A multiplier of 31 gives a VCO frequency of 221.932 MHz. Dividing that by 1 gives the output frequency 221.932 MHz, 0.88% higher than the target frequency.

Set N = 1. This produces a prescaler value of 1 + 1 = 2.

Set M = 30. This produces a VCO multiplier value of 30 + 1 = 31.

Set P = 0. This produces a postscaler value of 1.

Set C = 2, for a VCO frequency greater than 128 MHz.

A variety of programming values can be used. If the prescaler value is 3, this reduces f_{INTREF} to 4.773 MHz. A VCO multiplier of 46 would produce a VCO frequency of 219.545 MHz, just 0.21% less than the target frequency. With these values the final output frequency is closer to the target by nearly 0.7%, at the expense of a lower f_{INTREF}.

Set N = 2. This produces a prescaler value of 2 + 1 = 3.

Set M = 45. This produces a VCO multiplier value of 45 + 1 = 46.

Set P = 0. This produces a postscaler value of 1.

Set C = 2, for a VCO frequency greater than 128 MHz.

Additional Considerations for both PLL Styles

Dual PLL Constraint

For products with two PLLs, the two PLLs will interfere with each other (they will modulate the output frequency of the other PLL) if the higher frequency falls within 3 MHz of an integral multiple of the lower frequency. That is, if f_{higher} is the higher of the two frequencies and f_{lower} is the lower frequency, then the following equation must be satisfied:

$$f_{higher} \neq (n \times f_{lower}) \pm 3 \text{ MHz}; n = 1,2,3\ldots$$

Glitching

If the M or N programming value to the PLL is changed the PLL will transition smoothly from the original frequency to the new frequency.

However, if the P value is changed the output clock can "glitch".

For the RGB624 and RGB640 products, special circuitry has been added to the output of the SYSCLK PLL which allows "P" to change without glitching the clock output. A requirement of this circuitry is that a P value of 0 (divide-by-1) is illegal.

Programming Registers by Product

RGB514, RGB525

These products have a single, old style PLL. A bank of registers can be set up, to allow external frequency selection by choosing one register combination out of the set.

For direct programming, the Fixed PLL Reference Divider register at index 0x0014 is used to supply a single REF DIV COUNT, or N value, for all frequencies. A bank of sixteen registers labeled F0, F1, …F15, at index 0x0020 through 0x002f, holds the VCO DIV COUNT, or M value, and the DF bits. One of these registers is selected to provide the desired M value to the PLL.

For pseudo M/N programming the sixteen F0 - F15 registers are grouped into 8 pairs of M/DF, N values, and the Fixed PLL Reference Divider register is not used.

RGB524, RGB528A

These products have the same pixel PLL programming registers as the RGB514 and RGB525, and in additional there is a single pair of registers for programming the second, SYSCLK PLL.

The new registers are called System PLL Reference Divider (index 0x0015, for the N value), and System PLL VCO Divider (index 0x0016, for the M value and the DF bits.)

The Fixed PLL Reference Divider at index 0x0014 is now called the Fixed Pixel PLL Reference Divider.

RGB526/RGB526DB, RGB624/ RGB624DB

The RGB526, RGB526DB, RGB624 and RGB624DB have two new style PLLs. For backwards compatibility with the RGB51x and RGB52x products, the System PLL Reference Divider, System PLL VCO Divider, Pixel Fixed PLL Reference Divider, and F0 - F15 registers are retained for use with direct or pseudo M/N programming.

For standard M/N programming of the new style PLLs, the existing PLL registers are reconfigured, and new registers are added.

For the SYSCLK PLL, the register at index 0x0015 becomes the N value and index 0x0016 holds the M value. A new register at 0x0017 holds the P value, and a new register at index 0x0018 holds the C value.

For the Pixel PLL the sixteen registers at indices 0x0020 - 0x002f are redefined as four sets of four values: M, N, P, and C.

RGB561

This product has a single, old style PLL. The PLL Reference register at index 0x0022 provides the N value. The PLL/VCO Divider register at index 0x0021 provides the M value and PFR bits (PLL Frequency Range, or DF bits.)

RGB640

This product has two new style PLLs. The registers at indices 0x0010, 0x0011, 0x0012 and 0x0013 provide the N, M, P and C values, respectively, for the Video (Pixel) PLL. The registers at indices 0x0014, 0x0015, 0x0016 and 0x0017 provide the N, M, P and C values, respectively, for the Auxiliary (SYSCLK) PLL.

Data Sheet is Final Authority

This application note tries to illustrate how to program any of the RGB Palette DAC products. However, in cases where a data sheet differs from this application note, the data sheet is the official specification for the operation of the product.

Table 7. Summary of Changes

Date	Changes
05/08/95	1. First publication.
05/29/97	1. Deleted reference to operation faster than 250 MHz. 2. Deleted references to Palette DACs which are not marketed. Added references to Palette DACs which were introduced following the original publication of this application note. 3. Updated Table 2 with current data sheet document numbers. 4. Fixed formatting errors in Figure 3.

Printed in the United States of America
5-97

IBM Microelectronics Division
1580 Route 52, Bldg. 504
Hopewell Junction, NY
12533-6531

The IBM home page can be found at:
http://www.ibm.com

The IBM Microelectronics Division home page can be found at:
http://www.chips.ibm.com

The IBM Palette DAC web pages can be found at:

http://www.chips.ibm.com/products/display

SK10-3010-01