A Generic Architecture for Multi-Modulus Dividers in Low-Power and High-Speed Frequency Synthesis

Raja K. K. R. Sandireddy, Foster F. Dai, Richard C. Jaeger

Dept. of Electrical and Computer Engineering, Auburn University, Auburn, AL-36849 USA

Abstract — This paper presents a generic architecture for programmable multi-modulus dividers (MMD) for low-power and high-speed frequency synthesis applications. The proposed architecture uses cascaded divide by 2/3 cells in a ripple fashion except for the last cell, which is a P/P+1 dual modulus prescaler used to adjust the minimum division ratio and the required division range. This approach provides an optimized architecture with minimum current consumption, the smallest area and minimum number of control bits for designing MMDs with a unit step increment.

I. INTRODUCTION

Wireless communication has received increasing interest for military and commercial applications. With numerous wireless LAN (WLAN) standards operating in different frequency bands, market leading WLAN solutions must offer multi-mode interoperability with transparent worldwide usage. The demands for fast switching and high operating frequencies make the design of frequency synthesizers a challenging task [1]-[2]. The synthesizer can be an integer-N type with programmable integer frequency dividers or Fractional-N type synthesizer. However, for multi-standard applications, it is often difficult to cover multiple frequency bands using an integer frequency synthesizer whose step size is limited by the reference frequency. In order to achieve fine step size to cover the multi-band channel frequencies, one has to lower the reference frequency in an integer-N synthesizer design, which results in a high division ratio for the PLL and thus high in-band phase noise. In contrast, fractional-N synthesizers allow the PLL to operate with a high reference frequency and also achieve fine step size by constantly swapping the loop division ratio between integer numbers, thus on an average dividing by a fractional number. A fractional-N synthesizer can include either a multi-modulus divider or dual modulus prescalers. A dual modulus prescaler can be implemented easily, but it cannot synthesize all the required frequencies. Its programmability is further limited if a delta-sigma noise shaper is designed in the fractional-N synthesizer, where the instantaneous division ratio varies around the average correct division ratio. Therefore, an MMD is highly desired in delta-sigma fractional-N frequency synthesis.

In this paper, we propose a generic MMD algorithm along with the implementation of each cell which takes the minimum hardware and current. There have been several architectures proposed for MMDs [3]-[5]. It will be seen that the architecture given in [3] is a special case of the generic algorithm presented in this paper. When the MMD division range is not large compared to its minimum division ratio, the generic algorithm points out that the optimal MMD architecture is different from the one given in [3]. For the case of large MMD division range, the generic algorithm leads to the architecture with all divide by 2/3 cells which is the case in [3]. The architectures in [4] and [5] use flip flops to implement the 2/3 cell, thereby consuming more hardware.

II. GENERIC MMD ARCHITECTURE

The proposed MMD architecture includes number of divide by 2/3 dual-modulus cells cascaded with a divide by P/P+1 dual-modulus cell (P being an integer and P 2) in a ripple fashion as shown in Fig. 1. In Fig.1, only the last cell is selected to be a divide by P/P+1 cell so that all the division ratios in the required range can be programmed with unit step increment. If any of the preceding cells is not a 2/3 cell, then the unit step increment is not guaranteed. If a step increment other than one is desired, the optimal architecture is to place a fixed ratio divide by S stage in front of the MMD cells as shown, so that the

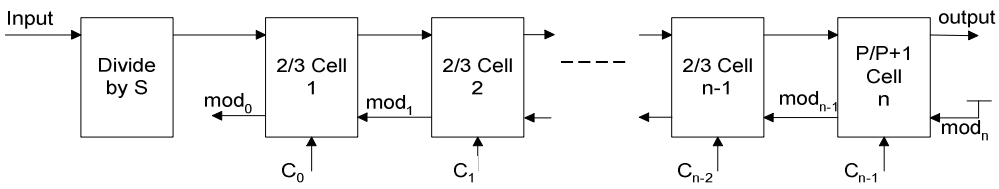


Fig. 1. A generic MMD architecture.

 0-7803-8703-1/04/$20.00 © 2004 IEEE

MMD has a programmable step size of S. Without losing generality, we will assume that the MMD discussed in this paper will have a unit step size (S=1). Thus, the architecture shown in Fig.1 provides the *output* signal period as given by the equation:

$$T_{output}=(2^{n-1}P+2^{n-1}C_{n-1}+2^{n-2}C_{n-2}+\ldots+2^{1}C_{1}+C_{0})T_{input}$$

where T_{output} and T_{input} are the *output* and *input* periods respectively and C_0, C_1, … C_{n-2}, C_{n-1} are the programmable MMD control bits given to the cells as shown in Fig. 1. It can be observed from the previous equation that the last cell, being P/P+1, increases the minimum division ratio maintaining the unit step increment for the MMD.

To determine the number of cells needed and to select a proper division ratio of *P*, the following generic algorithm shall be followed:

1. *Assume that the required division ratio is from D_{min} to D_{max}; the division ratio range is $(D_{max} - D_{min} + 1)$.*
2. *If the required range is greater than the minimum division ratio, D_{min}, the MMD is referred to the architecture in [3].*
3. *The implemented MMD's range, defined from M to N, can be larger than the required range. Initially set $M = D_{min}$.*
4. *Now the number of cells required becomes $n=\lceil \log_2(D_{\max} - M + 1)\rceil$ where function $\lceil a \rceil$ denotes rounding a to the nearest integer towards plus infinity.*
5. *The division ratio for the last cell can be found from $P=\lfloor M/2^{n-1}\rfloor$, where function $\lfloor a \rfloor$ denotes rounding a to the nearest integer towards zero.*
6. *If $M/2^{n-1}$ is not an integer, then reset $M = P*2^{n-1}$ and go to step 4.*
7. *If $M/2^{n-1}$ is an integer, we have to decide recursively whether using a single P/P+1 cell or using a combination of a 2/3 cell and a $\frac{\lfloor P/2 \rfloor}{\lfloor P/2 \rfloor+1}$ cell will achieve lower current consumption and smaller die size, as discussed later.*
8. *The final MMD architecture is thus a combination of stages with*

$$\left(2/3\right)_1 \quad \left(2/3\right)_2 \quad .... \quad \left(2/3\right)_{n-1} \quad \left(P/P+1\right)_n$$

If we are using all 2/3 cells then the total number of cells required is $\lceil \log_2(D_{\max}+1)\rceil - 1$.

Suppose we require division ratios from $D_{\min}$ = 102 to $D_{\max}$ = 108. Initially, the implemented MMD design range starts from M = 102 and the number of cells required is n = 3. Then the division ratio of the last cell becomes P = 25 and M is reset to 100. Next, the number of cells required is recalculated as n = 4, the division ratio of the last cell is 12, and M is set to 96. On the next iteration of the steps 4 to 6, the values of *n, P, M* remain the same. Since the implementation of a 12/13 cell requires less hardware and consumes less current than the combination of a 2/3 cell cascaded with 6/7 cell, the last cell division ratio is maintained at P = 12. Thus the final MMD architecture for this example is

$$\left(2/3\right) \quad \left(2/3\right) \quad \left(2/3\right) \quad \left(12/13\right).$$

If the required division ratio were from 68 to 108, the algorithm results in an MMD architecture with six 2/3 cells cascaded i.e. P = 2, which is the case presented in [3].

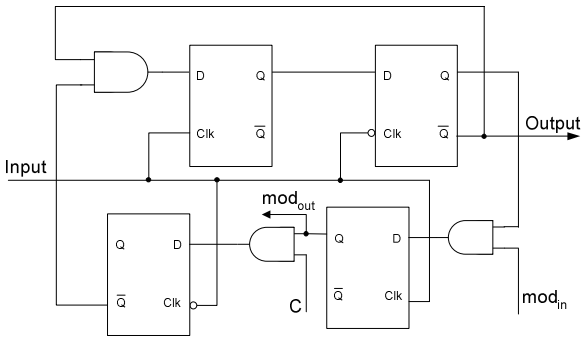


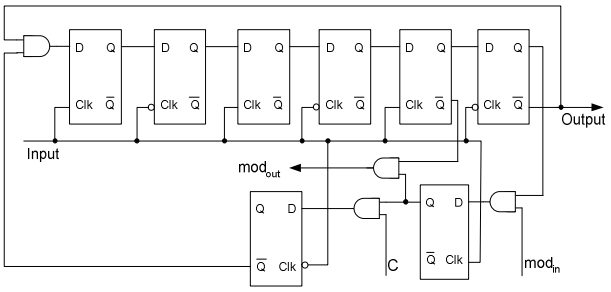
Fig. 2 Logical implementation of divide by 2/3 cell with mod control.

III. IMPLEMENTATION

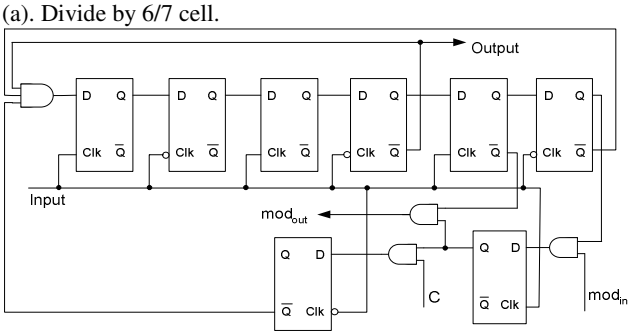
The divide by 2/3 cell is implemented as shown in Fig. 2 and a similar implementation is extended to any P/P+1 cell as explained later. The divide by 2/3 cell divides the input frequency by 2 or 3 depending on the inputs mod_{in} and C. The *output* period is twice that of the *input* unless mod_{in} = 1 and C = 1, for which the *output* period is three times the *input* time period. The other output signal mod_{out} has the same time period as *output* when mod_{in}=1, but with a different duty cycle. When connected in a ripple chain as shown in Fig. 1, the last cell always has its mod_{in} input as logic high, and for all other cells, the mod_{in} input is high for only one *input* cycle during one *output* cycle, adding an extra *input* cycle to the output. Thus, the instantaneous division ratio of that cell is set to be 3 provided its control input C = 1.

The divide by 2/3 cell logic discussed above can be extended for any P/P+1 cell. However the logical implementation of the cell differs for P being an even number from that of P being an odd number. For instance, the logical implementation for divide by 6/7 and 5/6 cells is as shown in Fig. 3. The gate that has mod_{in} as its input can be avoided because the mod_{in} of the last cell is always logic high. So, the output of the last latch on the top row of latches can be directly connected to the D input of the right latch on the bottom row.

For high-speed operation, current-mode-logic (CML) is chosen to implement the divide by 2/3 cell and divide by P/P+1 cell. There are various architectures of the 2/3 divider cell such as the one in [5], where the proposed divider has 3 flip flops (one with reset). The resettable latch requires one more transistor level in CML implemen-



(a). Divide by 6/7 cell.



(b). Divide by 5/6 cell.

Fig. 3 Logical Implementation of divide by (a) 6/7, (b) 5/6 cells

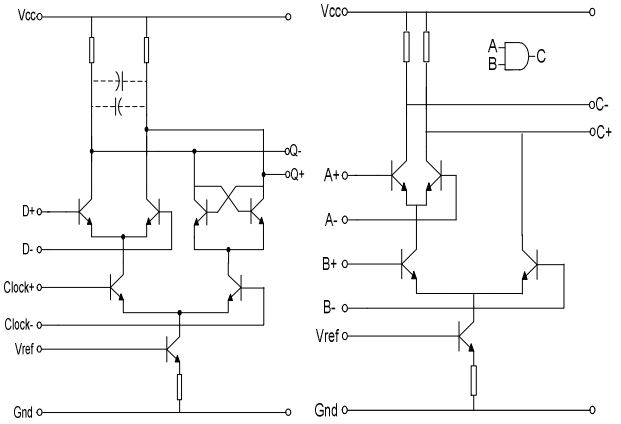


Fig. 4 Differential CML circuit implementations of D Latch and AND gate.

-tations, which is practically difficult when the supply voltage is low. Alternate implementations of resettable latches have been investigated, but it requires additional tail current for gating the input D of a latch [5]. Another drawback of the 2/3 circuit presented in [5] is that a common signal, *strobe*, goes to all the latches and thus needs to be buffered, causing high power consumption. We therefore choose the implementation shown in Fig. 4 to implement the dual modulus prescalers. Fig. 4 shows the differential CML implementations of the D-latch and the AND gate required for 2/3 and P/P+1 cells. The capacitors connected using dotted lines are deglitch capacitors.

IV. Discussion on Current Consumption and Area

It can be observed from Fig. 4 that the CML latch and AND gate consume the same current if they operate at the same frequency, say, in the same dual modulus cell. In this paper, the SiGe 47 GHz f_T technology is used as a reference to calculate current consumption.

A switching circuit operating up to 1/3rd of the transistor f_T frequency has been reported. Considering the temperature, power supply and processing variations, we need to leave some margin for the design. Moreover, the switch circuit output often has glitches during signal transition. A simple solution to deglitch the output waveform is to place a couple of deglitch capacitors across the collector load as shown in Fig. 4. The deglitch capacitor filters the high frequency glitches, but it degrades the operation speed of a switching circuit. So, we need to bias the circuit slightly higher in order to reach the desired operation speed. As a rule of thumb, with deglitch capacitors, we can assume that the operation of a switching circuit is guaranteed up to 1/6th of the transistor f_T frequency. For instance, if we set the bias current as 1mA, which corresponds to transistor f_T = 45GHz, the resultant switching circuit can thus operate up to 7.5GHz. Transistor f_T curves can be used for this purpose. However, it's not recommended to reduce the bias current below 50 A for emitter lengths of 2.5 m. Too small bias currents tend to increase the switching circuit noise. Moreover, small bias current requires large load resistors to maintain the internal differential voltage swing of about 400mVpp. Large load resistor occupies large die size and increases the parasitic capacitance, which degrades the circuit speed. For low power applications, a smaller transistor size can be chosen. Smaller transistors tend to have higher 1/f noise, which will contribute to synthesizer in-band phase noise.

The layout of CML circuits is really a case-by-case analysis. For generality, we discuss here the MMD size issues based on a typical CML layout. The area required by the load resistors is about the same required by the transistors with emitter length of 2.5 m for 1 mA bias current. However, for latches operating at lower frequency, the bias current is scaled down and thus the load resistor value as well as the deglitch capacitance increases. The area required by the transistors remains almost the same, while the area required by the load resistors proportionally increases as the bias current decreases. For instance, if the bias were scaled down to 333 A, the area required by the resistors would be 3 times that required by the load resistors when the bias current was 1mA, in order to maintain the same internal voltage swing.

Consider an application with the VCO running at 7.5GHz that requires a division ratio from 28-35. Using the proposed algorithm, the MMD can be realized using the stages 2/3 → 2/3 → 7/8. The implementation of the 2/3 cell presented in this paper uses 4 latches and 3 gates. So, the first stage requires a total of 7mA for the 4 latches and 3 gates since the bias current for each latch or gate at

7.5GHz is 1mA. The second stage requires 2.8mA since bias current for each latch or gate is 0.4mA. The third stage which is a 7/8 cell uses 10 latches and 3 gates and is biased at 0.2mA for each latch or gate, thus requiring a current of 2.6mA for the stage. The three control bits which are buffered using a flip flop each require 6mA since these buffers operate at the highest frequency. Thus the total current consumption for this application is calculated as 18.4mA. The CML circuit implementation of a latch requires 7 transistors and 3 resistors of which 2 are load resistors while an AND gate requires 5 transistors and 3 resistors. At 1mA, each latch requires 21 times the transistor area, while each AND gate requires 15 times the transistor area. The area required for the first stage is equivalent to 129 times the transistor area, but for the second stage the load resistors size increases by 2.5 times in order to maintain the same voltage swing, and the area required for the second stage is 194 times the transistor area. The third stage requires an area equivalent to 595 times the transistor area. Thus the total area required by the implementation is 918 times the transistor area.

V. MMD ARCHITECTURE FOR WLAN APPLICATIONS:

Let's now apply the generic MMD algorithm to a WLAN transceiver design. Since synthesizers can have large power consumption and often use considerable area, it is very important to limit the number of components required by the radio architecture. From this perspective a direct down conversion architecture would be preferable to a classical super-heterodyne radio as there is only one synthesizer rather than two. Thus, for IEEE 802.11a mid/low band, $F_{CH} = F_{VCO} = 5180+20k, (k=0,1,2\ldots7)$. If a 40MHz reference source is used, the MMD division ratio needs to be $D = F_{VCO}/40 = 129 + (k+1)/2$, namely, $D = 129.5\sim133$. Assuming that a 2nd-order delta-sigma noise shaper with 2-bit output is used to remove the fractional spurs, the instantaneous MMD division ratio could vary from –1 to 2 around its average division ratios. Thus, the MMD division range for IEEE 802.11a mid/low bands is given by $D = 128\sim135$. Applying the Generic MMD algorithm discussed previously, the MMD can be realized using the cells

2/3→ 2/3→ 2/3→ 2/3→ 8/9.

The previous example for IEEE 802.11a application points out that the division range of a MMD for fractional-N synthesizer is often small with respect to its minimum division ratio. The results obtained using the discussion in the previous section is tabulated in Table 1. It can be observed that the generic MMD design algorithm presented in this paper provides a better architecture with less logic and less power consumption than the prior art MMD designs using straight divide by 2/3 cells [3],[4]. Output waveforms are shown for the division ratio of 135 of the IEEE 802.11a band in Fig. 5. The mod_{out} waveforms of all the cells have the same frequency with different duty cycles. The time period of the input signal, *in* is 10 ns, and correspondingly the outputs have period of 1350ns as shown. For clarity, only one cycle of the outputs has been shown.

	No of stages/ control bits	No of latch/ gate	Size in terms of transistor area (relative to this work)	Current consumption (relative to this work)
This work	5/5	36/15	3219 (1)	22.55mA (1)
Vaucher[3]	7/7	42/21	4272 (1.33)	26.95mA (1.2)
Chen[4]	7/7	56/21	5567 (1.73)	30.65mA (1.36)

Table 1 Comparison of different MMD architectures for IEEE 802.11a band

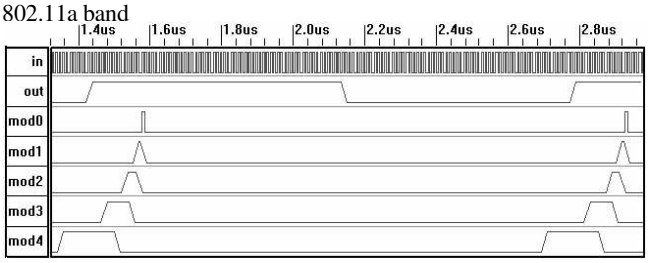


Fig. 5 Input and output waveforms along with the mod_{out} waveforms for the division ratio 135 of the IEEE 802.11a band.

VI. CONCLUSIONS

A generic MMD algorithm has been presented. It has been shown that the MMD implemented with this algorithm for the IEEE 802.11a mid/low band is superior to the prior art architectures. There has been a considerable improvement in area, current consumption, and number of control bits required by the MMD. This algorithm can be implemented for any range to obtain a low power MMD with minimum hardware.

VII. REFERENCES

[1] J. Craninckx and M. Steyaert, *Wireless CMOS Frequency Synthesizer Design,* Dordrecht, The Netherlands: Kluwer Academic Publishers, 1998.

[2] B. Razavi, "Challenges in the design of frequency synthesizers for wireless applications", *Proc IEEE Custom Integrated Circuits Conf., Santa Clara, CA, USA*, pp. 395-402, May 1997.

[3] Vaucher, C.S., Ferencic, I., Locher, M., Sedvallson, S.,Voegeli, U.,Wang, Z., "A family of low-power truly modular programmable dividers in standard 0.35- m CMOS technology," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 7, pp. 1039 -1045, July 2000.

[4] Wei-Zen Chen, Jieh-Tsorng Wu, "A 2-V, 1.8-GHz BJT phase-locked loop," *IEEE Journal of Solid-State Circuits*, vol.: 34, no. 6, pp. 784 -789, June 1999.

[5] Sheng, N.-H., Pierson, R.L., Wang, K.C., Nubling, R.B., Asbeck, P.M., Chang, M.-C.F., Edwards, W.L., Philips, D.E., "A high-speed multimodulus HBT prescaler for frequency synthesizer applications," *IEEE Journal of Solid-State Circuits*, vol. 26, no. 10, pp. 1362 -1367, October 1991.