A 5.3 GHz Programmable Divider for HiPerLAN in 0.25μm CMOS

N. Krishnapura, P. Kinget
Bell Laboratories, Lucent Technologies,
Murray Hill, NJ, 07974, USA.

Abstract

A 5.3 GHz low voltage CMOS frequency divider whose modulus can be varied from 220 to 224 is presented. Programmability is achieved by switching between different output phases of a D flip-flop (DFF). An improved glitch-free phase switching technique through the use of a retimer circuit is introduced. A high speed low voltage DFF circuit is given. The programmable divider fabricated in $0.25\mu m$ technology occupies 0.09 mm^2 and consumes 24 mA at 1.8 V and 37 mA at 2.2 V. 5.5 GHz operation with 300mV$_{pk}$ single ended input is achieved with a 2.2 V supply. The residual phase noise at the output is -130 dBc/Hz at an offset of 1 kHz from the carrier.

1. Introduction

HiPerLAN is a wireless data networking standard that operates in the 5.3 GHz band and consists of 5 channels separated by 23.5294 MHz. The phase locked loop used for synthesizing the carrier uses a frequency divider whose division factor is to be programmable from 220 to 224 and whose input is at 5.3 GHz.

In the following section, we discuss phase switching, which is the chosen architecture for this programmable divider. In section 3 we discuss a potential problem with existing phase switching circuits and introduce a retimer block which eliminates this problem. Section 4 deals with the high speed divide-by-2 ($\div 2$) stage. Section 5 discusses the overall circuit and some implementation details. Experimental results are given in section 6.

2. Architecture

Conventionally, programmable dividers are implemented using a high speed dual modulus prescaler along with low speed programmable counters which implement arbitrary division factors by "swallowing" pulses[1]. In [2], an asynchronous divider is presented in which pulse swallowing is accomplished by switching between different output phases of a $\div 2$ stage implemented using a master-slave flip-flop. In this scheme, only one flip-flop in the first $\div 2$ stage operates at full-speed (f_{in}) instead of several flip-flops as in a dual-modulus prescaler. Additionally, the absence of high speed feedback loops around multiple flip-flops as present in a dual modulus prescaler and the reduced load on the VCO result in a higher maximum speed of operation and a lower power consumption in a given technology. This architecture was therefore chosen for our 5.3 GHz programmable divider.

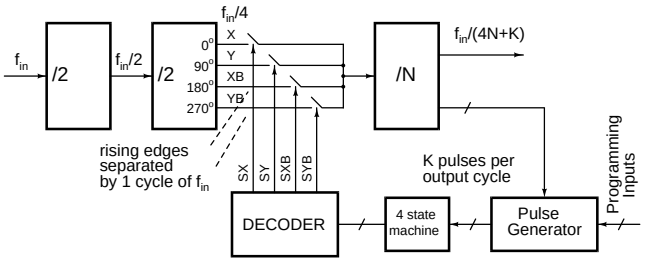

Figure 1. Programmable frequency divider employing phase switching.

Fig. 1 illustrates the principle of operation. A quarter cycle (90^o) at $f_{in}/4$ is the same duration as a full cycle (360^o) at f_{in}. A cascade of two $\div 2$ stages using master-slave flip-flops has four outputs (X, Y, XB, YB) which are separated by 90^o (Fig. 1, top four waveforms in Fig. 2). At any instant, only one of these outputs is connected to the subsequent $\div N$ stage through a multiplexer (MUX). In order to swallow a cycle and augment the total count of the frequency divider by 1, the input of the $\div N$ stage is switched to a waveform that is lagging the current waveform by 90^o (e.g from X to Y in Fig. 1). For an arbitrary division factor, input cycles can be swallowed by changing the control inputs of the MUX appropriately. In the absence of phase switching, the divider chain has a division factor of $4N$. If the phases are switched K times in each cycle of the output of the divider chain, K input cycles are swallowed and consequently, the division factor is augmented by K and becomes $4N + K$. By varying K, programmability is achieved. The pulse generator block (Fig. 1) generates K pulses per output cycle where K is set by the programming inputs. A $\div 4$ counter (Fig. 1) can be used as a 4state machine that is clocked by these pulses and cycles through four states. Each state corresponds to one of the four possible connections in the MUX. A decoder decodes the state and turns the appropriate switch ON (through one of SX, SY, SXB, SYB) in the MUX.

3. Glitch free phase switching

A potential problem with the architecture of Fig. 1 is shown in Fig. 2. The input clock and the four waveforms (X, Y, XB, YB) differing by 90^o are shown. Shown below these are two possible output waveforms. The *ideal* one is the re-

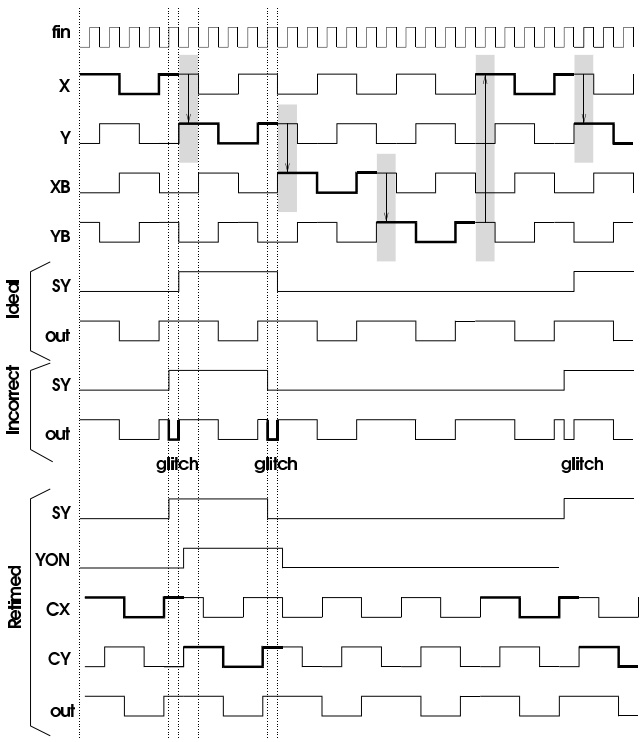

Figure 2. (Incorrect): Occurrence of a glitch during phase switching due to improper signal timing, (Retimed) Elimination of glitches by retiming of MUX input signals.

sult of switching from X to Y controlled by a low-high transition in SY at the correct instant. Switching at the *incorrect* instant causes a glitch which introduces an extra transition and results in a miscount of the synthesizer. From these waveforms it is clear that the switching from wave-X to wave-Y must occur when both X and Y are high. The timing window for glitch-free switching is shown shaded in the top part of Fig. 2 for each of the four transitions (marked with arrows). The circuit controlling the MUX must ensure the correct timing.

If no precautions are taken, glitches are almost certain to occur because

1. The delay of the divider and the logic circuit in the feedback path that provides the phase switching signals cannot be determined accurately over process and temperature variations.

2. As mentioned earlier, X must be changed to Y only when X·Y=1. Y must be changed to XB only when XB·Y=1 and so on (see the shaded regions in Fig. 2). So the correct "windows" are different for each of the transitions in the MUX.

In a previous implementation glitches were avoided by using a long rise-time for the MUX control signals [2] which isn't a robust solution or by using feedback from the output of the MUX[3] which limits the speed due to the delay in the feedback loop. Here we describe a retimer circuit (Fig. 3) that takes as its input the four clocks X, XB, Y, YB and the

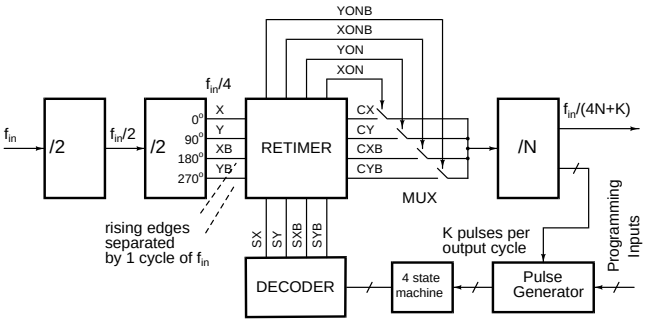

Figure 3. Programmable divider employing glitch free phase switching.

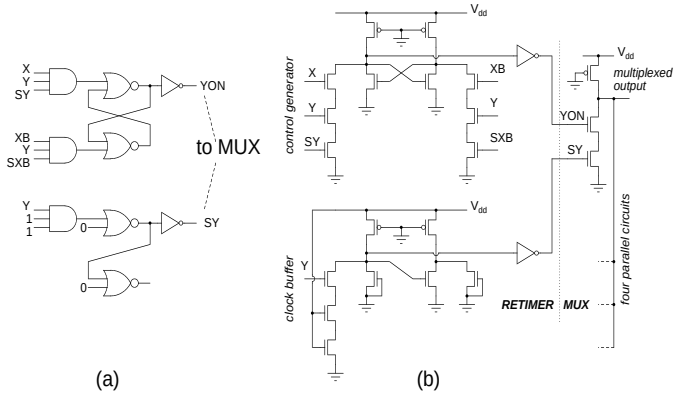

Figure 4. (a) Retiming circuit used to achieve glitch free phase switching, (b) pseudo-nMOS implementation

control signals SX, SXB, SY, SYB (decoder outputs). The retimer generates four clocks CX, CY, CXB, CYB and corresponding control signals XON, YON, XONB, YONB whose transitions are synchronized to the respective clocks so that the possibility of glitches is eliminated. These signals drive the MUX. Consistent with the high speed requirement, the retimer operates in a feedforward fashion as described below.

The logic diagram and a pseudo-nMOS implementation of the retimer are shown in Fig. 4. The output of a cross coupled latch (YON) directs Y to the output of the MUX. YON is pulled high if both X and Y are high and SY is high (i.e. the decoder selects Y). YON is pulled low when both Y and XB are high and SXB is high (the decoder selects XB). In order to have both the clocks (CX, CY, CXB, CYB) and the control signals (XON, YON, XONB, YONB) arrive synchronously at the MUX inputs, a buffer whose delay is the same as that of the control signal generator is used on the clock line (lower part of Fig. 4). Its operation is illustrated with the bottom waveforms in Fig. 2. SY goes high when Y is still low. But the retimer's outputs YON and CY are synchronized to each other, thus avoiding a glitch. For each of the other three clocks, identical circuits are used which generate the control signal for a particular clock in the proper "window".

From simulations, this circuit is found to work reliably with variations in process and temperature. The arrival of the clocks and the control signals at the inputs of the MUX is found to be practically coincident.

4. Divide by two stage

The block most difficult to design is the first $\div 2$ stage which should operate at 5.3 GHz or more. Both a conventional latch and a single phase latch[4] are too slow for our purposes because they have a large input capacitance due to the parallel connection of pMOS and nMOS gates. Due to its lower mobility and larger threshold voltage (in our case, $V_{TN} = 0.6V$ and $V_{TP} = 0.9V$), the pMOS transistor contributes little to the current drive and a lot to the capacitances, and considerably slows down the circuit.

The latch proposed in [5] uses pMOS transistors in the clock path and was found to work only up to ≈ 2 GHz in our technology. Also, the 25% duty cycle of the output signals is inconvenient for phase switching. The source coupled latch (e.g. [2]) has a reduced output swing which facilitates high speed, but due to stacking of many devices it cannot be accommodated in a low supply voltage.

Using pseudo-nMOS gates enables a high speed operation while providing a large output swing. For comparison, we observe that in this technology, with a 1.8 V supply, a 3 stage CMOS ring oscillator oscillates at 2.5 GHz whereas a 3 stage pseudo-nMOS ring oscillator oscillates at 6 GHz.

Fig. 5 shows a pseudo-nMOS DFF whose outputs are connected back to its inputs (shown in dashed lines) to form a $\div 2$ stage. NAND gates are used to form the latch since they enable a compact layout where node parasitics can be minimized. In order to have the inputs vary around the switching threshold of the gates, CLK & CLKB are ac coupled through 0.2 pF capacitors. An inverter whose input and output are tied together biases the DFF inputs to the correct dc level over process and temperature. A disable mode in which the inputs to the $\div 2$ stage are tied to the opposite rails prevents self oscillations. From simulations, the circuit (extracted from layout) was found to operate satisfactorily over process and temperature variations with 300 mV peak inputs (single ended) at 5.5 GHz input while driving the second $\div 2$ stage.

5. The complete programmable divider

Fig. 3 shows the block diagram of the complete programmable divider. The division factor in absence of phase switching is 216 ($= 4N = 2^3 \times 3^3$). The $\div 2$ stages use the circuit shown in Fig. 5. Successive stages use smaller transistors, since they are operating at a reduced speed. Pseudo-nMOS inverters are used for interstage buffering in the divider chain.

A $\div 3$ stage needs two DFFs, one of which has AND gated inputs. Gating can be achieved simply by adding parallel branches appropriately to the input of the DFF shown in Fig. 5. To realize division factors from 220 to 224, four to

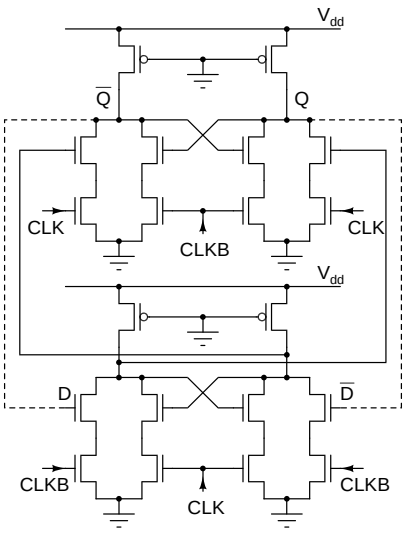

Figure 5. pseudo-nMOS divide by two stage.

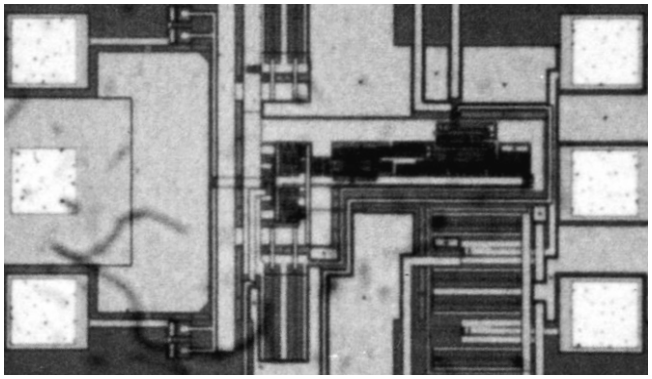

Figure 6. Chip photograph.

eight pulses are required and they are derived from the outputs of the last three stages. Combinational logic is used to obtain the desired number of pulses from the 3 programming bits. The retimer circuit is shown in Fig. 4. The 4State counter, decoder and MUX shown in Fig. 3 are pseudo-nMOS implementations. For the sake of measurement, a string of CMOS inverters is used to buffer the output. It can provide $0.6\,V_{pp}$ across a 50Ω load. Broadband matching for the clock inputs is provided by on chip 50Ω termination resistors.

Fig. 6 shows the photograph of the chip. The entire test chip measures 0.9mm^2 out of which the programmable divider (excluding the buffers) takes 0.09 mm^2. Separate supply lines are used for the output buffers. On chip bypassing is provided using MOS capacitors.

6. Measurement Results

The chips were tested by probing on wafer as well as by wirebonding the dice to a printed circuit board. The differential inputs were provided using a hybrid. The output was monitored on an oscilloscope and a spectrum analyzer.

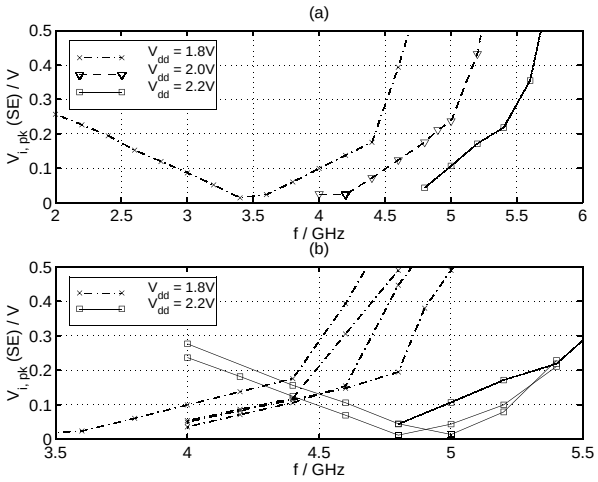

Figure 7. (a) Divider sensitivity vs. input frequency at different supply voltages, (b) Different samples with V_{dd}= 1.8 V and 2.2 V.

Fig. 7(a) shows the minimum input amplitude required for proper operation with three supply voltages (1.8 V, 2.0 V, 2.2 V). The curves are qualitatively similar, but shifted to the right with increasing supply voltage. Assuming 300 mV peak (single ended) to be a reasonable upper limit of the drive available from an on chip VCO, the corresponding upper limit on the frequency can be ascertained from Fig. 7. 5.5 GHz operation is possible with a supply voltage of 2.2 V. 5.5 GHz operation at 1.8 V (with a 300 mV peak input) was expected from simulations over process and temperature variations, but was not achieved because some of the process parameters of the experimental process shifted severely after submission of the layout. Measurement of several chips across the wafer showed a consistent performance as can be seen from Fig. 7(b). Consistent performance over all the possible division factors was also verified.

The current drawn from the supply varies slightly with input frequency and division factor. The divider draws a maximum of 24 mA, 30 mA and 37 mA respectively with supply voltages 1.8 V, 2.0 V and 2.2 V.

For residual phase noise measurement, two dividers were run from a common input clock. The outputs were fed to the phase detector in a HP3048A phase noise analyzer. The measured noise is twice the noise of each of the dividers and the jitter in the input clock is canceled. Details of this measurement technique can be found in [6]. Fig. 8 shows the results of the measurement. The residual phase noise in the output is shown. The sum of the residual phase noise of the two dividers at 1 kHz offset is -130 dBc/Hz. As can be seen from Fig. 8, 10 dB/decade behavior is maintained down to 1 Hz. The absence of $1/f^3$ regions in the residual phase noise is good for the overall synthesizer since the divider contributes to the synthesizer's jitter only at low offset frequencies that are inside the PLL's bandwidth.

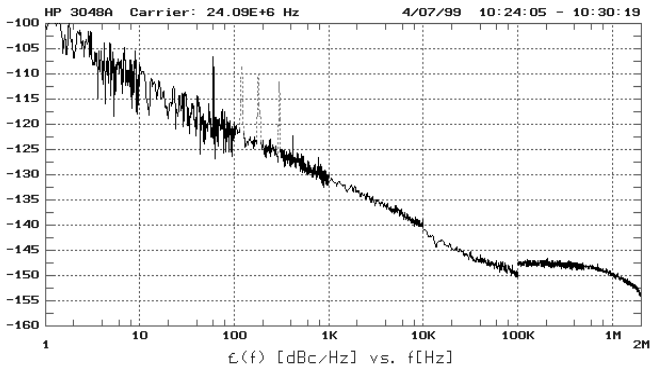

Figure 8. Sum of the residual phase noise from two dividers.

7. Conclusions

A high speed programmable divider operating at a low supply voltage is presented. A DFF capable of high speed operation is described. A retiming technique for reliable implementation of phase switching programmable division is given. Measurement results demonstrate the high speed capability and programmability of the presented circuits. Low noise performance compared to other published circuits is achieved despite the high frequency and low voltage operation.

8. Acknowledgments

We are very grateful to V. Boccuzzi for the excellent help with the measurements. We wish to acknowledge A. Dunlop's support and the loan of equipment by M. Banu, R. C. Melville and H. Wang.

[1] U. Rohde, *Digital PLL Frequency Synthesizers: Theory and Design*, Englewood Cliffs: Prentice Hall, 1983.

[2] J. Craninckx and M. S. J. Steyaert, "A 1.75-GHz/3-V Dual-Modulus Divide-by-128/129 Prescaler in 0.7-μm CMOS", *IEEE Journal of Solid State Circuits*, vol. 31, no. 7, pp. 890-897, July 1996.

[3] M. H. Perrott, *Techniques for High Data Rate Modulation and Low Power Operation of Fractional-N Frequency Synthesizers*, PhD thesis, Massachusetts Institute of Technology, Sep. 1997.

[4] J. Yuan and C. Svensson, "High-Speed CMOS Circuit Technique", *IEEE Journal of Solid-State Circuits*, vol. 24, no. 1, pp. 62-70, Feb. 1989.

[5] B. Razavi et al., "Design of High-Speed, Low-Power Frequency Dividers and Phase Locked Loops in Deep Submicron CMOS", *IEEE Journal of Solid State Circuits*, vol. 30, no. 2, pp. 101-109, Feb. 1995.

[6] W. F. Egan, "Modeling Phase Noise in Frequency Dividers", *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, vol. 37, no. 4, pp. 307-315, Jul. 1990.