

High-Speed Architecture for a Programmable Frequency Divider and a Dual-Modulus Prescaler

Patrik Larsson

Abstract—We present a prescaler architecture that is suitable for high-speed CMOS applications. We apply the architecture to a 4/5 and an 8/9 dual-modulus prescaler and obtain a measured maximum clock frequency of 1.90 GHz in a standard 0.8 μm CMOS bulk process. This is 13% faster than the traditional prescaler architecture keeping the same power consumption. We also apply the key part of the prescaler to a divide-by- N circuit reaching 1.75 GHz. This is three times faster than any previously reported CMOS implementation and comparable to GaAs implementations.

I. INTRODUCTION

ONE of the high-frequency building blocks in a communication system is a frequency synthesizer including a frequency divider. Traditionally, the divider has been realized with a high-speed technology such as bipolar or GaAs. CMOS is the cheapest technology today and seems to be the cheapest alternative for the foreseen future, indicating that CMOS implementations are advantageous.

There are three types of frequency dividers: cascaded divide-by-two stages, dual-modulus prescaler, and programmable divider (also called divide-by- N circuit). Many applications require a programmable division ratio, excluding the use of divide-by-two stages. In this paper, we present a novel architecture for both a dual-modulus prescaler and a divide-by- N circuit.

Some CMOS prescalers have been presented [1]–[3] with a maximum operating frequency not far behind bipolar [4], [5] or GaAs [6], [7] implementations. See [5], [7], and [8] for further references. These circuits have used advanced processing and/or special circuit techniques in combination with a standard frequency divider architecture. In this paper we focus on improving the architecture, and we compare our results with the traditional architecture using the same processing technology and circuit technique.

II. DUAL-MODULUS PRESCALER

All prescalers that we have found are based on a shift register ring with logic similar to Fig. 1(a). This is an 8/9 prescaler [2], which easily can be modified to a 4/5 prescaler [1], [6]. Our approach is to preprocess the

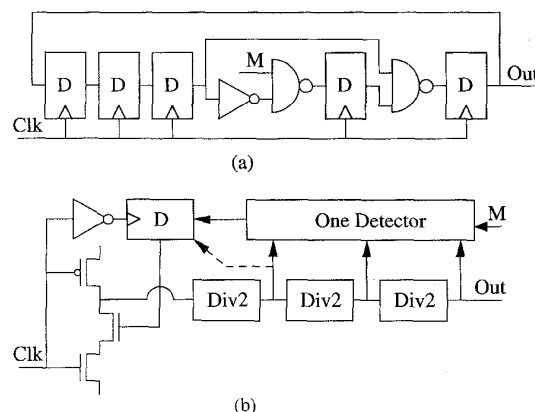


Fig. 1. Dual-modulus 8/9 prescalers. (a) Based on a shift register ring. (b) With preprocessing of the clock signal.

clock signal and then use cascaded divide-by-two stages as sketched in Fig. 1(b). The one detector gives a low output pulse when the M signal and the outputs of all divide-by-two stages are high. This pulse is delayed one clock cycle and synchronized with the inverted Clk signal in the D flip-flop. A low output will prohibit one negative pulse of the Clk signal from reaching the first divide-by-two stage. After canceling one clock pulse, it will take eight clock pulses before the detector output goes low, giving a division ratio of nine. By setting M low, the detector output is always high and the prescaler divides by eight. The circuit operates as an 8/9 prescaler, but can easily be modified to a $2^{b/2^b} + 1$ prescaler by adding/removing divide-by-two stages and extending/shortening the detector.

Since the rightmost divide-by-two stage reaches one before the other stages, the one detector can combine early arriving information first [9], and the critical path in the detector is the detection of the leftmost bit in the divider chain. In principle, this bit should be sent to the detector as in Fig. 1(b). To enhance speed, the detection of the leftmost bit was done with very simple logic inside the D flip-flop as the dashed line indicates. Since the D flip-flop has a small load of only one transistor, it has roughly the same speed as the first divide-by-two stage. This indicates that the maximum operating frequency of the clock preprocessing prescaler is nearly the same as that of a chain of cascaded divide-by-two stages. The following divide-by-two stages operate at a lower frequency than that of the first stage, which makes it possible to use smaller transistor sizes reducing the load of the first stage. Therefore,

Manuscript received May 4, 1995; revised August 26, 1995.

The author was with the Department of Physics and Measurement Technology, Linköping University, S-581 83 Linköping, Sweden. He is now with AT&T Bell Laboratories, Holmdel, NJ 07733 USA.

Publisher Item Identifier S 0018-9200(96)03397-5.

transistor sizing is an efficient method for speed enhancement and power reduction of this architecture.

The gating of the clock signal in Fig. 1(b) utilizes dynamic logic since the input of the first divide-by-two stage is floating when *Clk* is high and the output of the *D* flip-flop low. If the circuit is to be clocked at very low speed, dynamic logic is not suitable. We can simply eliminate the dynamic mode by adding a *P* transistor driven by the output of the *D* flip-flop, i.e., the gating of the clock signal is done with a *Nand* gate instead of the three-transistor structure in Fig. 1(b).

We implemented the shift register ring in Fig. 1(a) and the clock preprocessing structure in Fig. 1(b) for both an 8/9 and a 4/5 prescaler. We used the same circuit technique for both prescalers. All transistor widths were limited to 20 μm except for some transistors in the clock buffers and for a few clocked transistors in the ring structure that was sized larger. The prescalers were implemented in a standard 0.8 μm CMOS process for which relevant process parameters are given in Table I. The inverter delay was measured in a five-stage ring oscillator which is characterized in Fig. 2(a). The inverter delay was smaller for a three-stage oscillator with identical delay stages as seen in Fig. 2(b). We believe this is caused by the internal nodes not having full swing in the three-stage oscillator. Each delay stage was an inverter with a 100 μm wide *P*-transistor and a 52 μm wide *N*-transistor with no sharing of source and drain diffusion areas.

Table II shows the measurement results from the two prescaler structures, and their measured maximum operating frequency are plotted as function of power supply voltage for two different chips in Fig. 3(a) and (b). The prescalers work properly for both division ratios at the maximum operating frequency. The maximum operating frequency is comparable to that of [1]–[3] taking into account the different process technologies and flip-flop implementations. To avoid high-speed I/O signals, a divide-by-16 circuit was added at the output of the prescalers. This was implemented with the same dynamic circuit technique as the prescalers. Its minimum output frequency was measured to be lower than 100 kHz, and we conclude that the ring structure can be clocked at a frequency of 100 kHz. In the clock preprocessing architecture, the last stage is clocked at a much lower frequency than the input clock leading to a minimum input frequency of $2^b \times 100$ kHz, where 2^b is the division ratio. The minimum clock frequency limit can be alleviated by using static CMOS for the MSB's operating at lower speed than the high-speed dynamic LSB stages. The figures for power consumption in Table II do not include the clock power that is considerably larger for the shift register ring than the clock preprocessing architecture. Simulations of extracted layout predicted a power consumption of 22–28 $\mu\text{W}/\text{MHz}$ for the prescalers including the clock power. We only give approximate results of power consumption since noise coupling between internal VCO's on the chip reduced the accuracy of the power measurements as explained in the following. Several different types of divid-

TABLE I
CMOS PROCESS PARAMETERS

Power supply	V _{dd} = 5V
Minimum gate length	L = 0.8 μm
Gate oxide thickness	t _{ox} = 16nm
Measured inverter delay at V _{dd} =5V	t _{inv} = 114ps

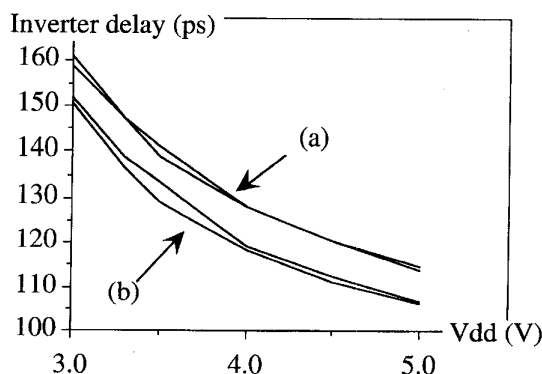


Fig. 2. Measured inverter delay in ring oscillator for two different chips as function of power supply voltage. (a) Five-stage and (b) three-stage oscillator.

ers were laid out on a single chip sharing a single power supply. Each divider had its own individually controlled VCO. The only way of estimating the power consumption of a specific divider was to measure the total power consumption as function of the input frequency of that divider and assume that the power consumption is linearly proportional to the input frequency as is common for CMOS circuits. However, when changing the speed of the VCO feeding the divider for which power consumption was to be estimated, noise coupled from the VCO to other VCO's changing the input frequencies of the other dividers slightly. Therefore, the change in power consumption was not only due to the frequency change of the divider for which we intended to measure the power consumption.

III. AN ASYNCHRONOUS DIVIDE-BY-*N* CIRCUIT

A programmable frequency divider is often built from a counter with load or reset and a detection circuit as in Fig. 4. By using a backward counter, detecting the 0 state and loading the counter with the division ratio, the propagation delay of the detector can be made small [9]. By realizing that the higher order bits (MSB's) of the counter reach the 0 state before the lower order bits (LSB's), we can combine early arriving information first and we only need quick detection of the LSB's. Since the MSB's reach zero earlier than the LSB's, we can also let the toggling of MSB's lag behind the LSB's [10]. Therefore, an asynchronous ripple counter consisting of cascaded programmable divide-by-two stages is sufficient. This avoids the carry propagation that is limiting the speed of synchronous counters. The dividing function is also obtained with a forward counter, but then the counter is loaded with the one's complement of the binary word that represents

TABLE II
MEASURED PRESCALER CHARACTERISTICS

Type of prescaler	Division ratio	Speed (GHz)	P/f @ Vdd=5V ($\mu\text{W}/\text{MHz}$)	Area (μm^2)
Shift register ring	4/5	1.68	$\approx 12\text{-}16^*$	70x150
[1, 2 and others]	8/9	1.68	$\approx 12\text{-}16^*$	100x140
Clock preprocessing	4/5	1.90	$\approx 16\text{-}20^*$	70x140
[This work]	8/9	1.90	$\approx 16\text{-}20^*$	80x160

* Approximate measurement excluding input clock power.

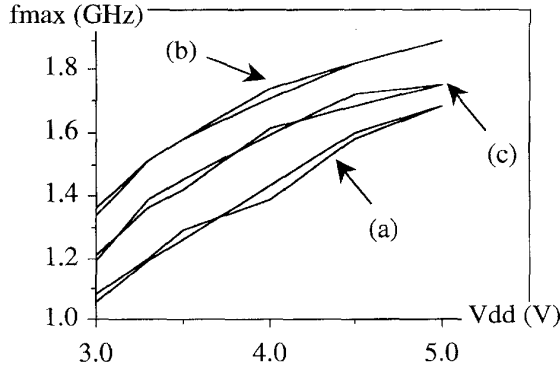


Fig. 3. Measured maximum operating frequency as function of power supply voltage. (a) Traditional dual-modulus prescaler in Fig. 1(a). (b) Dual-modulus prescaler in Fig. 1(b). (c) Divide-by-N circuit in Fig. 7.

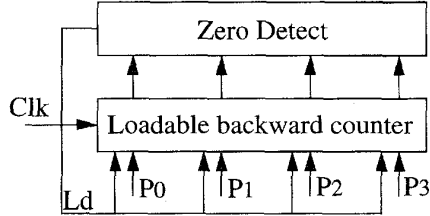


Fig. 4. Programmable frequency divider based on a loadable counter.

the division ratio. In the following we will discuss a forward counter that detects the 11...11 state to perform the loading.

A problem for an asynchronous counter is the loading. In a forward counter, the divide-by-two stages consist of a negative edge-triggered flip-flop. After detecting the 1 state, the counter is loaded with a word that contains some zeros. If bit i is loaded with a zero, bit $i + 1$ will toggle since it triggers on negative edge. Therefore, the load signal needs to be applied to all bits simultaneously to prohibit some bits from toggling. Thus, the load signal is global and has high fanout indicating a speed bottleneck.

We remove the speed limiting requirement of having a global load signal with a simple idea; instead of loading some bits of the counter with zeros from the 1 state, we let the counter go one step beyond 11...11, which is 00...00. From this state we load some bits with ones. Loading bit i with a 1 will not toggle bit $i + 1$. With this principle, the load signal need not be global, and it can

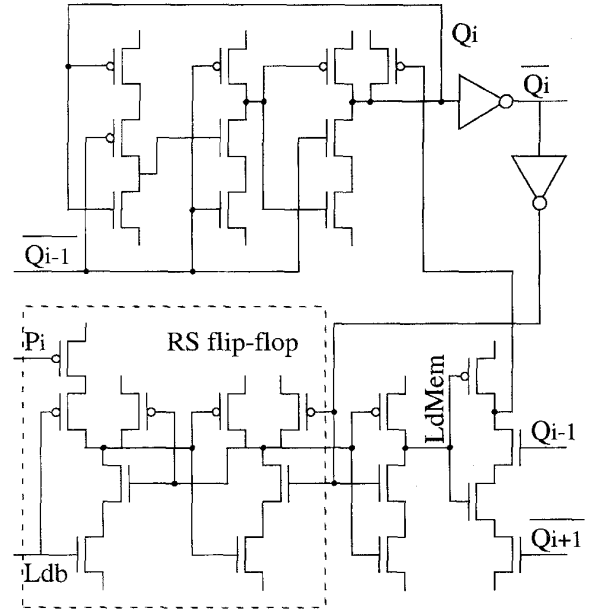


Fig. 5. Transistor schematic of a single bitcell in a counter with Ripple Inhibited Load.

be buffered as it propagates along the bitslices. First, we create the load signal when all bits reach one. Then, the load signal is propagated along the divider to set a memory bit, LdMem, in each bitcell as indicated in Fig. 5. When the counter is switching from 11...11 to 00...00, there will be a ripple signal through all divide-by-two stages. Immediately when a bit has switched from one to zero and the LdMem bit is set, the output is returned to one and LdMem is reset. We call this principle ripple inhibited load. If bit i is to be loaded with a zero, no loading is necessary and therefore the LdMem bit is only set for those bits that are to be loaded with a one. This is realized with the P transistor driven by P_i in Fig. 5. The top half of Fig. 5 is a loadable positive edge-triggered divide-by-two circuit followed by a static inverter.

P_i is the i th bit of the binary word that determines the division ratio. If P_i is low, bit i should be loaded with a one since ones complement of P should be loaded into the counter. If P_i is low and Ldb falls, the LdMem node will go high and it stays high even if Ldb returns to high. When Q_i falls, the RS flip-flop is reset, but LdMem is latched at the value loaded by the Ldb signal. To make sure that Q_i

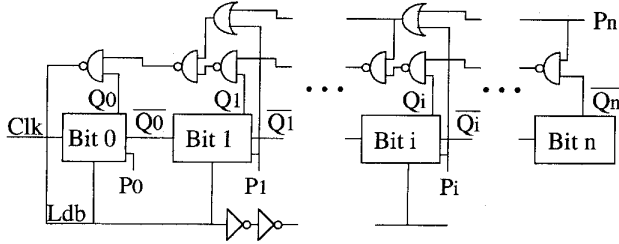


Fig. 6. Block schematic of our divider based on a counter with Ripple Inhibited Load.

stays low long enough to toggle the next bit, we introduce a simple handshaking protocol with bit $i + 1$. This is realized by the N -transistor that is driven by the inverse of Q_{i+1} . Bit i is not allowed to load itself with a one until Q_{i+1} has been falling, i.e., the inverse of Q_{i+1} is high.

To simplify the load operation of the toggle stage of bit i , we do not perform the loading until the clock signal of bit i is low, i.e., Q_{i-1} is high. Since we know that the clock signal is low, the loading of bit i can be performed with a single P transistor that pulls up the output of bit i . Some weak transistors were added to the bitcell in Fig. 5 to make sure that noise will not corrupt charge storage on dynamic nodes. This makes the divider very safe at the cost of reduced maximum clock frequency.

The block schematic of our asynchronous programmable divider is shown in Fig. 6. If the division ratio is a small number, e.g., 010...00, the ripple of (unimportant) higher order bits when going from state 11...11 to 00...00 might corrupt the next detection of the 11...11 state. Therefore, we preprocess the binary word P . If all bits of P to the right of position i are zero, the output of the 1 state detection circuit of bit i (the leftmost NAND gate) is set to high i.e., the real values of Q_j , $j \geq i$ are irrelevant. All bitcells are identical to the transistor schematic in Fig. 5 except for the last stage, which is a simple divide-by-two circuit without load. The reason why the last bit does not need any load is that we combine its output with the P_n value in a NAND gate before sending it to the 1 state detection chain. If P_n is zero, we can ignore the actual value of Q_n and if P_n is high, it should be loaded with a zero (ones complement of one) which need not be done since Q_n is zero after the 11...11 to 00...00 transition. Since the load signal does not need to be a global signal applied to all bitslices simultaneously, the Ldb signal is buffered after two bitslices. A more detailed description of this divider can be found in [8].

IV. COMBINING THE PRESCALER AND THE DIVIDE-BY- N CIRCUIT

When examining our prescaler in Fig. 1(b), we see that it consists of a state detection circuit, a ripple chain, and a preprocessing circuit. The ripple chain is a fixed chain without load. By replacing the ripple chain and the state detection circuit with the asynchronous divider presented in the previous section, we get a new divide-by- N circuit

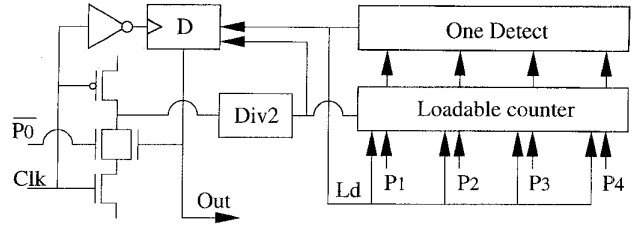


Fig. 7. A divide-by- N circuit with preprocessing of the clock signal.

drawn in Fig. 7, which is shown to have considerably higher maximum operating frequency than the asynchronous divider by itself. The output frequency of the preprocessing unit in Fig. 7 is at most half of the input frequency, so the preprocessing unit can be clocked at twice the frequency of the asynchronous divider. The basic operation is as follows. If the division ratio is an odd number ($P_0 = 1$), we cancel one input clock pulse when the load signal is activated. After canceling one pulse, there is an even number of clock pulses before the load signal is activated the next time. Therefore, we can divide the incoming clock signal by two before feeding the asynchronous divider. If the division ratio is an even number ($P_0 = 0$), we divide the incoming clock by two without canceling any clock pulse. Note the irregular connection of P_0 in Fig. 7. This is the least significant bit of the binary word that represents the division ratio and it determines whether to cancel a clock pulse or not when the load signal is active.

We used the 0.8 μm CMOS process characterized in Table I when designing a six-bit frequency divider. We measured a maximum operating frequency of 1.75 GHz at $V_{dd} = 5$ V as indicated in Fig. 3(c), which shows the maximum operating frequency as function of the power supply voltage. The minimum frequency was the same as for the prescalers discussed above. Previously published works are summarized in Table III. Note that the programmable divider here is running faster than the traditional prescalers based on a shift register ring listed in Table II.

Reference [10] is a maximally pipelined synchronous divider with minor transistor sizing leaving most transistors 4 μm wide. The maximum operating frequencies of our clock preprocessing divider and the dividers presented in [10] and [11] are independent of the number of bits in the dividers, while the structures in [7] and [9] have a large fan-out that will be troublesome to drive for dividers with many bits. As for the dual-modulus prescalers, noise coupling degraded the accuracy of the power measurements, so we give only approximate values of the power consumption.

V. CONCLUSIONS

Most high-speed prescalers and frequency dividers that have been reported use an advanced process and/or circuit technique to reach high operating frequency. We focus on architectural developments and introduce a new dual-

TABLE III
PERFORMANCE OF DIVIDE-BY- N CIRCUITS

Ref.	Technology	Vdd	Ratio	Bits	Speed (MHz)	P @ fmax (mW)	Area/bit (μm^2)
[7]	0.7 μm GaAs	1.0V	≥ 2	4	1700	36	?
[7]	0.7 μm GaAs	1.0V	≥ 2	6	1400	90	127000
[9]	1 μm GaAs	3, -3V	≥ 3	6	1500	1400	275000
[10]	1 μm CMOS	5V	≥ 2	6	400	80	7600
[11]	2 μm BiCMOS	4.5V	≥ 8	15	$\geq 165^*$	55	25000
[12]	0.8 μm CMOS	5V	≥ 3	14	500	8	16000
This work	0.8 μm CMOS	5V	≥ 2	6	1750	50-60**	6900

* Simulated with worst case speed parameters.

** Approximate measurements excluding clock power.

modulus prescaler architecture. In a standard 0.8 μm CMOS bulk process, we measured a maximum operating frequency of 1.90 GHz for our architecture, compared to 1.68 GHz for the traditional prescaler architecture implemented in the same process.

We combine the key part of the prescaler with a novel asynchronous divider and reach 1.75 GHz in a standard 0.8 μm CMOS process. This is three times faster than any previously reported CMOS or BiCMOS divide-by- N circuit and is similar to several reported GaAs circuits.

Dual-modulus prescalers were invented because of the inferior speed of divide-by- N circuits. With our architecture, we show that there is only a small difference in maximum operating frequency between a dual-modulus prescaler and a divide-by- N circuit, indicating that dual-modulus prescalers might be obsolete in the future.

REFERENCES

- [1] H. I. Cong, J. M. Andrews, D. M. Boulton, S. C. Fang, S. J. Hillenius, and J. A. Michejda, "Multigigahertz CMOS dual-modulus prescaler IC," *IEEE J. Solid-State Circuits*, vol. 23, pp. 1189-1194, Oct. 1988.
- [2] R. Rogenmoser, Q. Huang, and F. Piazza, "1.57 GHz asynchronous and 1.4 GHz dual-modulus 1.2 μm CMOS prescalers," in *Proc. Custom Integrated Circuit Conf.*, '94, 1994, pp. 16.3.1-4.
- [3] N. Foroudi and T. A. Kwasniewski, "CMOS high-speed dual-modulus frequency divider for RF frequency synthesis," *IEEE J. Solid-State Circuits*, vol. 30, pp. 93-100, Feb. 1995.
- [4] T. Aytur and B. Razavi, "A 2GHz, 6mW BiCMOS frequency synthesizer," in *Proc. International Solid-State Circuit Conf.*, 1995, pp. 264-265.
- [5] T. Seneff, L. McKay, K. Sakamoto, and N. Tracht, "A sub-1 mA 1.5-GHz silicon bipolar dual modulus prescaler," *IEEE J. Solid-State Circuits*, vol. SC-29, pp. 1206-1211, Oct. 1994.
- [6] M. Ohhata, T. Takada, M. Ino, N. Kato, and M. Ida, "A 4.5-GHz GaAs dual-modulus prescaler IC," *IEEE T. Microwave Theory Tech.*, vol. 36, pp. 158-160, Jan. 1988.
- [7] H. P. Singh, R. A. Sadler, W. J. Tanis, and A. N. Schenberg, "GaAs Prescalers and counters for fast-settling frequency synthesizers," *IEEE J. Solid-State Circuits*, vol. 25, pp. 239-245, Feb. 1990.
- [8] P. Larsson, "Analog Phenomena in Digital Circuits," Ph.D. dissertation no. 376, Linköping Univ., 1995.
- [9] M. G. Kane, P. Y. Chan, S. S. Cherensky, and D. C. Fowls, "A 1.5-GHz programmable divide-by- N GaAs counter," *IEEE J. Solid-State Circuits*, vol. 23, pp. 480-484, April 1988.
- [10] P. Larsson, "A wide-range programmable high-speed CMOS frequency divider," in *Proc. IEEE International Symp. Circuits and Systems*, 1995, pp. 195-198.
- [11] C. S. Choy, C. Y. Ho, G. Lunn, B. Lin, and G. Fung, "A BiCMOS programmable frequency divider," *IEEE T. Circuits and Systems-II*, vol. 39, pp. 147-154, Mar. 1992.
- [12] J. C. Wu and H. H. Chang, "A 550 MHz 9.3 mW CMOS frequency divider," in *Proc. IEEE Int. Symp. Circuits and Systems*, 1995, pp. 199-202.