Circuits Syst Signal Process
DOI 10.1007/s00034-011-9279-8

A Gigahertz Digital CMOS Divide-by-*N* Frequency Divider Based on a State Look-Ahead Structure

Saleh Abdel-Hafeez · Ann Gordon-Ross

Received: 2 June 2010 / Revised: 3 February 2011

Abstract We present a scalable high-speed divide-by-*N* frequency divider using only basic digital CMOS circuits. The divider achieves high-speed operation using a novel parallel counter and a pipelined architecture. The parallel counter is based on a state look-ahead component in conjunction with an internal pipeline structure in order to simultaneously trigger all state value updates without a rippling effect. The pipeline latencies are precluded due to the use of a subtractor circuit that "swallows" any additional cycles. Furthermore, our frequency divider is easily scalable to large divider widths due to the use of modular component architecture. The fan-in and fan-out are independent of the divider width, thus making the structure attractive for regular VLSI implementation and continued technology scaling. We implemented our proposed divider using a 0.15-μm TSMC digital cell library and achieved a maximum operating frequency of 2 GHz, an area of 112 848 μm^2 (900 transistors), and consumed 15.47 mW of power operating at 2 GHz for an 8-bit design, which offers 252 different frequency divisions.

Keywords Divide-by-*N* · Frequency divider · High-speed · Look-ahead · Modulus · Parallel counter · Pipeline · Wide-range · 0.15-μm TSMC

S. Abdel-Hafeez (✉)
Jordan University of Science and Technology, Irbid 22110, Jordan
e-mail: sabdel_99@yahoo.com

A. Gordon-Ross
Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32011, USA
e-mail: ann@chrec.org

A. Gordon-Ross
NSF Center for High-Performance Reconfigurable Computing (CHREC), University of Florida, Gainesville, USA

Published online: 01 March 2011

1 Introduction

Frequency dividers are widely considered as a major limiting factor in frequency synthesizer systems, which require a very fast settling frequency feedback loop and a wide-range of frequency division ratios [12, 40, 42, 44]. Among the most important parameters of high-speed dividers are the operating frequency, operating range, and power consumption [31, 41, 42]. Most modern frequency dividers are typically classified as cascaded asynchronous programmable prescaler counters, programmable swallow counters, or programmable divide-by-N counters.

The cascaded asynchronous programmable prescaler counter [2, 6, 11, 20, 21, 23, 29, 33, 61, 62] offers a high operating speed due to the absence of a long delay loop (the feedback delay is only present between adjacent cells). However, the operating frequency is degraded by the increase in the number of cascaded cells, which are usually controlled by a synchronous binary programmable value. Furthermore, these counters suffer from a large accumulated jitter due to a long asynchronous cascaded cell topology.

The programmable swallow counter [26, 37, 52] provides a wide variety of dividing ratios, which makes this counter amenable to multi-system applications. These counters are composed of an N-bit programmable counter, an S-bit swallow counter, and a P-bit prescaler counter, which results in an input frequency (F_{in}) to output frequency (F_{out}) relationship of $F_{out} = (NP + S)F_{in}$. The structure has a minimum division ratio of P^2, meaning that it requires a delay of P^2 cycles to produce the output pulse. Furthermore, the structure lacks modularity, which limits its flexibility and reusability of low-cost design cells.

The divide-by-N frequency divider [1, 7, 28, 30, 46], also known as an integer-N divider, exploits a large wide-range division ratio which can vary from 2 to N where N is an arbitrary integer value less than 2^M for an M-bit divider. Even though the division ratio is not as flexible as a programmable swallow counter and the operation is not as fast as an asynchronous prescaler divider, research has shown that the integer-N frequency divider is more practical in terms of design time, continued cost-effective technology scaling, and has low spurious sideband effects compared to a fractional-N (i.e. swallow and prescaler) frequency divider [34, 54, 60].

Even though expensive technologies, such as BICMOS, GaAs, and SOI [8, 16, 25, 38, 50], can increase a divider's operating speed, CMOS is considered as one of the cheapest technologies today and will likely be the most advantageous alternative for future high-speed applications and power-efficient circuits. Therefore, many circuit optimization techniques [3, 9, 10, 13–15, 17, 19, 22, 24, 27, 35, 43, 45, 47, 49, 51, 53, 55, 57–59] are based on a CMOS differential topology in the form of analog and dynamic structures as a replacement for digital circuits, which are involved in designing programmable counters.

In this paper, we propose a gigahertz, wide-range, cost-effective technology scaling programmable divide-by-N frequency divider implemented in 0.15-μm CMOS technology. The key novelty of our frequency divider is its architectural improvements as compared to previous work. Since we implement all circuit blocks as digital logic CMOS structures, these circuit blocks can easily be replaced with alternative optimized dividers [3, 13, 15, 22, 35, 43, 49, 51] in order to maximize the operating

speed and further reduce the power consumption. The parallel operation of our frequency divider makes the architecture amenable to arbitrary width scaling, resulting in both high-speed and predictable operation. Our divider provides a low-cost implementation for continued technology scaling, which is attractive for both SOC and VLSI implementation. This paper is structured as follows. Section 2 presents our proposed programmable divide-by-N frequency divider architecture, with a new zero-latency pipeline end-of-count detector circuit (*EOC*) in Sect. 2.2 and a high-speed reload circuit with complete timing in Sect. 2.3. Section 2.4 presents our proposed wide-range parallel counter architecture based on a novel state look-ahead topology. In Sect. 3, we present experimental results and Sect. 4 discusses our conclusions.

2 Programmable Divide-by-N Frequency Divider Architecture

2.1 Background

Figure 1 depicts a block diagram for a conventional divide-by-N frequency divider consisting of a ripple counter, a reload circuit, an end-of-count detector (*EOC*) and external programmable inputs for frequency selection (*Frequency Control Variables*). During operation, the circuit begins in a reset state with an initial counter value equal to $N-1$. For each input clock pulse (F_{in}), the counter decrements and the output clock pulse (F_{out}) are generated when the counter value reaches "0". The *EOC* detects the "0" counter value and asserts *Frequency Match*, which signals the *Reload Circuit* (via *Restart*) to reset the *Ripple Counter* to $N-1$. This operation results in an output frequency of $F_{out} = F_{in}/N$, such that for an M-bit counter size, the frequency dividing factor N can be varied from 2 to $2^M - 1$. A variation of this divider counts up from "0" to a divider value N (instead of counting down to "0") (e.g., Chang and Wu [7]), which reduces the counter reload time to a single clock cycle (as opposed to the two clock cycles required when counting down) and increases the operating speed. The single-cycle reload process is accomplished by inserting

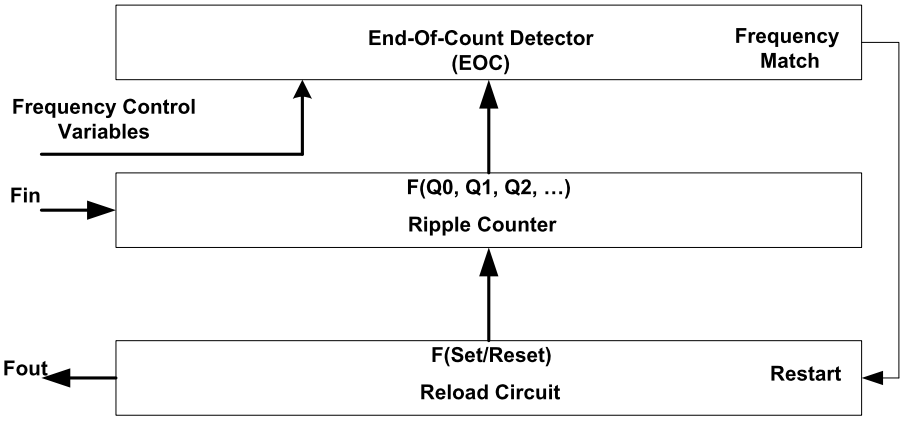

Fig. 1 Block diagram of a conventional divide-by-N frequency divider

a D-Type Flip-Flop (*DFF*) between the *EOC* circuit and the *Restart* signal that enables the *Reload Circuit*. Additionally, this architecture reduces input clock frequency constraints and increases divider operating frequency. However, this architecture has several drawbacks: the maximum operating frequency decreases as the size of the counter increases due to the increased delay caused by the ripple counter and the reload detector circuit is dependent on the counter size, which is usually implemented as a large cascaded connection of AND gates with a pipeline latency of one clock cycle.

Lee and Park [30] modified this counting-up architecture by combining the *Reload Circuit* and *EOC* circuit into one component operating as a control logic block. The control logic block increases operating frequency by dividing the reload process into three clock cycles, triggered by a count value of "2" instead of "0". Using this mechanism, they showed high-frequency operation regardless of the divider size. Kuo and Wu [26] noted that the control logic must be properly initialized during each reload in order to avoid circuit failure and alleviated this issue using an external signal, which was ORed with the restart signal to ensure the reload signal is active high, at the cost of losing the first clock cycle after every reload. However, the control block architecture is an irregular asynchronous structure that is unsuitable for wide-range programmable counters and these frequency dividers suffer from pipeline latency as well as long delays associated with the ripple counter and accumulated jitter.

2.2 Proposed Architecture

A characteristic feature of modern Phase-Locked-Loop (PLL) and frequency synthesizers for a system-on-chip (SOC) is a high-speed divider that can track the different values of the voltage-controlled oscillator (VCO) output signal and lock the feedback loop synthesizer dynamically. Our design has the following characteristics:

1. High-speed operation with a variety of programmable dividing frequency ratios.
2. Reconfigurable wide-scale dividing ratios with parallel operation that is independent of the divider size.
3. All design components are locally interconnected with a maximum fan-in and fan-out of four and three, respectively, independent of the divider size.
4. Cost-effective for continued technology scaling with applicability to a wide range of SOC design environments.
5. Fast design to market due to a well defined mathematical structure with simple digital logic components.
6. Easily optimized to a higher-order performance using high-cost technology or high-speed design components.

Figure 2 depicts a block diagram of our proposed divide-by-N frequency divider for a sample 8-bit divider (i.e., $M = 8$ resulting in 252 counting states). The circuit consists of novel parallel up-counter architecture and an improved *EOC* circuit combined with the reload logic. The parallel counter increases operating frequency by enabling all counting states to be simultaneously updated without suffering from the ripple delay introduced by ripple counters. This parallel operation activates the *EOC*

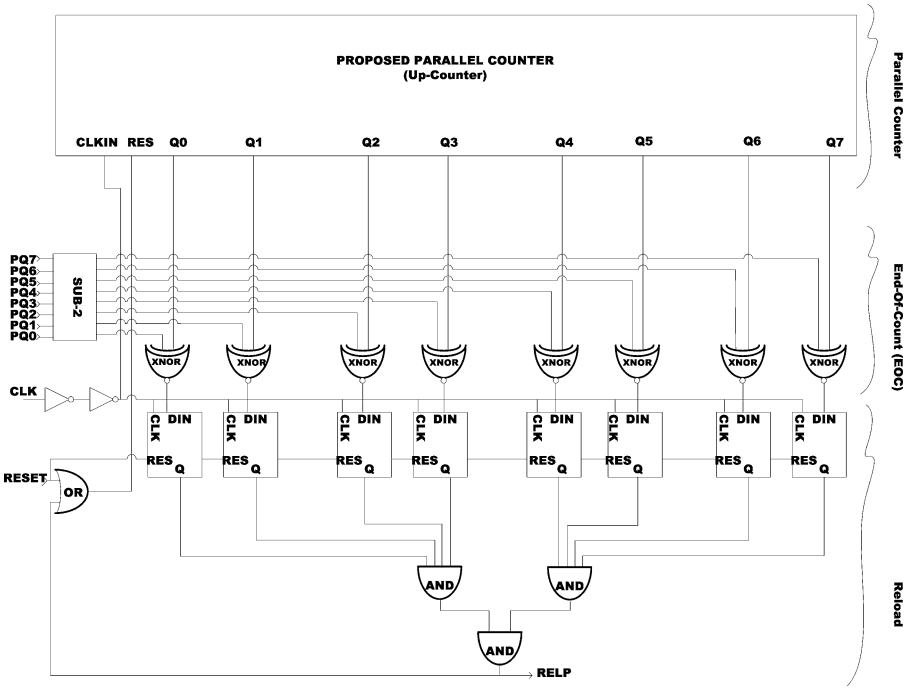

Fig. 2 Block diagram of our proposed divide-by-N frequency divider for a sample 8-bit divider

circuit with a uniform delay regardless of the counter size. The number of 2-input XNOR gates required by the *EOC* (G_{EOC}) is:

$$G_{EOC} = M \tag{1}$$

where M is the counter size.

The *EOC* circuit is activated when the counter value reaches $N-2$ using the *SUB*-2 logic block, which subtracts the programmable divisor value N by "2". The programmable divisor value N is specified using the divider size $M = 8$-bit with input ($PQ7$ to $PQ0$) as follows:

$$N = \sum_{i=0}^{M=7} PQ_i * 2^i \tag{2}$$

SUB-2 logic block is comprised of two cascaded *SUB*-1 logic blocks. Figure 3 depicts the *SUB*-1 logic block circuit for the sample 8-bit divider in Fig. 1. The *SUB*-1 logic block is an OR and XNOR gate structure that modularizes the inputs into groups of four (i.e., Fig. 3 has two groups of four for a sample 8-bit divider), which constrains the maximum fan-in and fan-out to four, provides predictable delay and total gate count for any arbitrary divider size, and is suitable and attractive for VLSI implementation. The total *SUB*-2 logic block delay has no effect on the divider timing since the *SUB*-2 logic can be set during an initial reset phase or during the reload process.

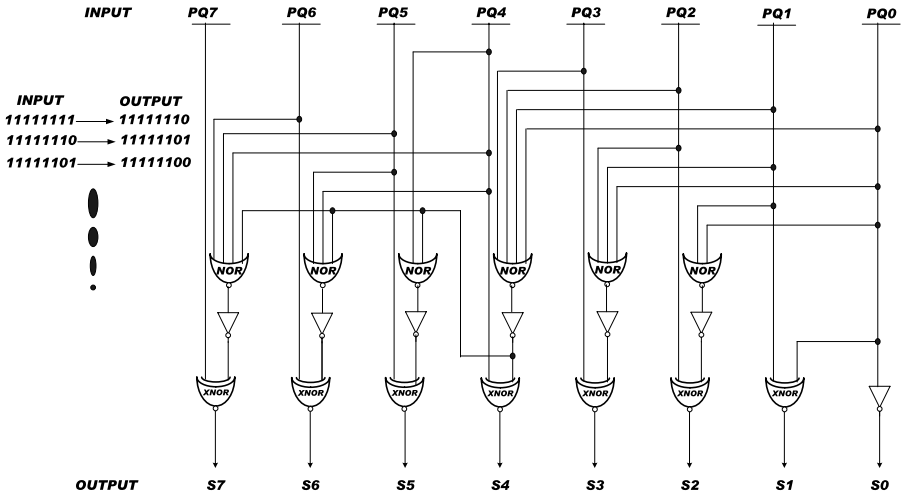

Fig. 3 SUB-1 logic block circuit for the sample 8-bit divider in Fig. 2

The reload circuit uses a pipelined *DFF* structure to divide the single-cycle operation into two clock cycles: one clock cycle to activate the *EOC* and the second clock cycle to activate the reload pulse (*RELP*) (similarly to *Restart* in Fig. 1) and thus reduces the accumulated delay of the *EOC* and *Reload* circuits. The *Reload* circuit is a tree structure of 4-input AND gates, where the total number of AND gates (G_{Reload}) for an M-bit frequency divider is

$$G_{Reload} = \sum_{i=1}^{\lfloor \mathrm{Log}_4(M) \rfloor} \left\lceil \frac{M}{4^i} \right\rceil. \tag{3}$$

The *RELP* generated by the *Reload* circuit re-initializes the parallel counter to "0" (reset state) with the rising edge of the system clock (*CLK*) and *RELP* is deactivated by *Restart* from the *EOC* through pipelined DFFs on the subsequent rising *CLK* with a single AND-gate delay. This method uses one complete cycle for resetting the parallel counter and a second cycle for deactivating the *RELP*. Since the divider uses a pipelined DFF structure, the output of the divider is generated after a two-clock-cycle latency. Therefore, in order to ensure there is no latency delay with respect to input frequency, *SUB*-2 subtracts a value of "2" instead of "1".

2.3 Timing Analysis

Figure 4 depicts the timing for our proposed divide-by-N frequency divider for the sample 8-bit divider in Fig. 2 showing the signal activities for detecting the *target value* "00011011" (i.e., $Q7 = Q6 = Q5 = Q2 = 0$ and $Q4 = Q3 = Q1 = Q0 = 1$) with a dividing factor $N = 27$ (i.e., every 27 clock cycles). We assume the parallel up-counter has been initialized to the starting count value of "00000000" and after the rising *CLK* edge, the counter state increments to the next state "00000001" and proceeds to count up after each rising *CLK* edge until the counter state equals

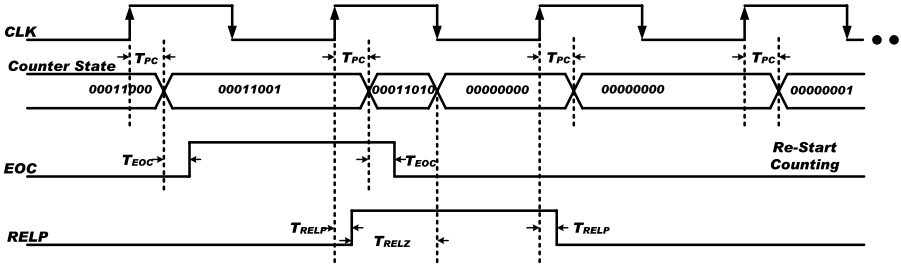

Fig. 4 Timing diagram of our proposed divide-by-N frequency divider for the sample 8-bit divider in Fig. 2 detecting the count value "00011011"

"00011000" (i.e., three states before the target value). On the rising edge of *CLK*, four events occur in sequence, assuming all DFFs are positive edge triggered:

Event 1) The counter state updates to "00011001" after a constant delay of T_{PC}, which is independent of the counter size.

Event 2) The *EOC* detects the counter state value after a delay of T_{EOC} such that:

$$T_{EOC} = T_{XNOR} + T_{SETUP\text{-}HOLD} \tag{4}$$

where T_{XNOR} is the delay of a 2-input XNOR gate and $T_{SETUP\text{-}HOLD}$ is the setup-hold time of the pipelined DFFs in the *Reload* circuit. Note that the detected counter state value by the *EOC* is actually $N-2$ due to the subtractor logic *SUB*-2 (Sect. 2.2).

Event 3) After the next rising *CLK* edge, *RELP* is activated after a delay of T_{RELP} such that:

$$T_{RELP} = T_{AND\text{-}TREE} + T_{PDFF\text{-}ACCESS} \tag{5}$$

where $T_{PDFF\text{-}ACCESS}$ is the delay of the pipelined DFFs and $T_{AND\text{-}TREE}$ is the AND-gate tree delay in the *Reload* circuit. Concurrently, the counter state updates to "00011010" after a delay of T_{PC}, the *EOC* is deactivated after a delay of T_{EOC}, and *RELP* is resetting the counter to "00000000" after a delay of T_{RELZ} such that:

$$T_{RELZ} = T_{RESET\text{-}COUNTER}. \tag{6}$$

Event 4) After the next rising *CLK* edge, *RELP* is deactivated after a delay of T_{RELP}, such that:

$$T_{RELP} = T_{AND\text{-}TREE} + T_{PDFF\text{-}ACCESS} \tag{7}$$

similarly to (5) (assuming falling/rising transient times of CMOS transistors are balanced).

In summary, the system clock period T_{CLK} can be defined based on the timing graph in Fig. 4 and a worst-case delay of one of the following paths:

$$T_{CLKIN} \geq (T_{PC} + T_{EOC}) = (T_{PC} + T_{XOR} + T_{SETUP\text{-}HOLD}) \tag{8}$$

$$T_{CLKIN} \geq (T_{RELP} + T_{RELZ}) = (T_{AND\text{-}TREE} + T_{PDFF\text{-}ACCESS} + T_{RESET\text{-}COUNTER}) \quad (9)$$

$$T_{CLKIN} \geq (T_{RELP}) = (T_{AND\text{-}TREE} + T_{PDFF\text{-}ACCESS}). \quad (10)$$

Equation (9) delay can be further reduced by inserting another pipelined DFF in the AND-tree logic gate path as part of the *Reload* circuit and modifying the subtractor log to subtract the value "3" instead of "2" in order to keep a zero pipeline latency. This modification would reduce $T_{AND\text{-}TREE}$ by one half.

2.4 Proposed Parallel Counter Circuit Architecture

Figure 5 depicts our proposed parallel counter architecture for a sample 8-bit counter. The main structure consists of the state look-ahead path (all logic encompassed by the dashed box) and the counting path (all logic outside the dashed box). *Module*-1 and *module*-3 are exclusive to the counting path and each module represents two counter bits. *Module*-1 is a standard parallel synchronous binary 2-bit counter, which is responsible for low-order bit counting and generating future states for all *module*-3*Ss* in the counting path by pipelining the enable signal for these future states through the state look-ahead path. Figure 6 depicts the (a) hardware schematic and (b) state diagram for *module*-1. *Module*-1 outputs $Q1Q0$ (the counter's two low-order bits) and $QEN = Q1$ AND $\overline{Q0}$ (the 1 in $QEN1$ denotes that this is the $QEN1$ for module-1). $QEN1$ connects to the module-2's *DIN* input.

Module-2 is a conventional positive edge-triggered *DFF* and is present in both paths. In the counting path, each *module*-3*S* is preceded by an associated *module*-2; hence, *module*-3*Ss* serve two main purposes. Their first purpose is to generate all counter bits associated with their ordered position and the second purpose is to enable (in conjunction with stimulus from the state look-ahead path) future states in subsequent *module*-3*Ss* (higher *S* values) in conjunction with stimulus from the state look-ahead path. Figure 7 depicts the (a) hardware schematic and (b) state diagram for

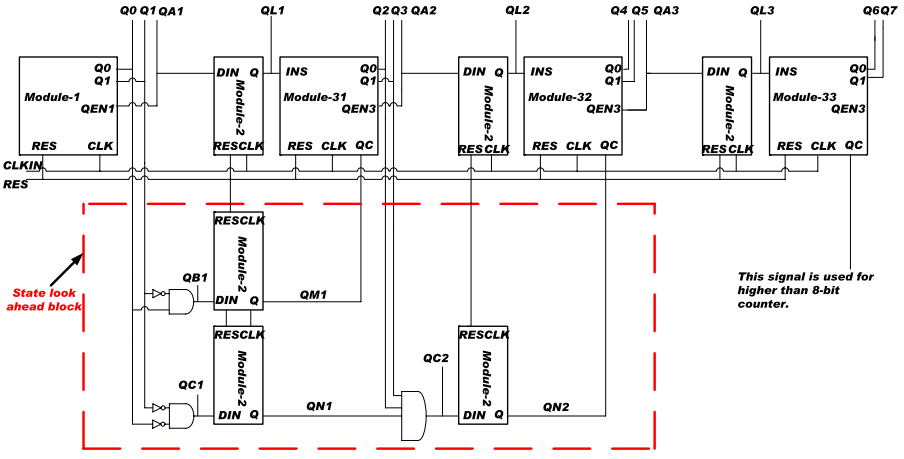

Fig. 5 A functional block diagram of our proposed 8-bit parallel counter with state look-ahead logic (encompassed by the *dashed box*) and counting logic (all logic outside the *dashed box*)

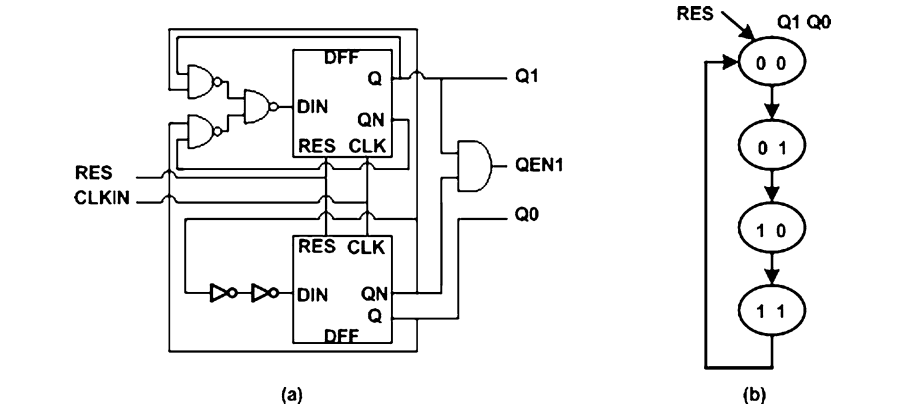

Fig. 6 Module-1 (**a**) hardware schematic and (**b**) state diagram

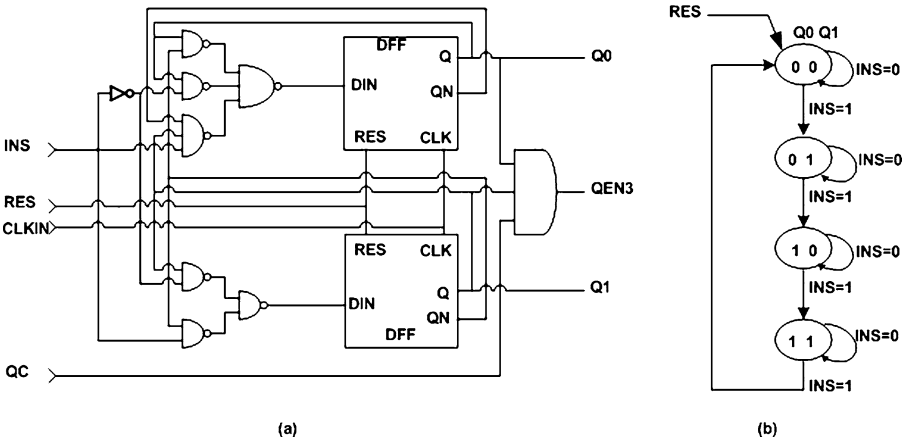

Fig. 7 Module-3S (**a**) hardware schematic and (**b**) state diagram

module-3S. *Module*-3S is a parallel synchronous binary 2-bit counter whose count is enabled by *INS*. *INS* connects to the Q output of the preceding *module*-2. *Module*-3S outputs $Q1Q0$ (which connect to the appropriate count output bits QX and $Q(X-1)$ as shown in Fig. 5) and $QEN3 = Q1$ AND $Q0$ AND QC (the 3 in $QEN3$ denotes that this is the QEN for *module*-3S). The state look-ahead logic provides the QC input (details discussed in Sect. 2.5). $QEN3$ connects to the subsequent *module*-2's *DIN* input and provides a 1-cycle look-ahead mechanism.

The novelty of the structure is summarized by having the next state transitions in counting modules of higher significance enabled on the clock cycle preceding the state transition using stimulus from the state look-ahead path. Subsequently, all counting modules concurrently transition to their next states at the rising clock edge (*CLKIN*). In such a way, the counting path's counting logic controls counting operations and the state look-ahead path's logic anticipates future states and thus prepares the counting path for these future states.

2.5 Counter's State Equations

To realize the structure in state equation form, we denote a past counter state using lower case $q7q6q5q4q3q2q1q0$ and the next counter state using upper case $Q7Q6Q5Q4Q3Q2Q1Q0$. The counter state equation necessary to enable $Q7Q6$ will contain $q5q4q3q2q1q0$. Consequently, $Q3Q2$ at *module*-31 is enabled by the past state $q1\overline{q0}$ from *module*-1, which carries through one clock cycle in the counting path's *module*-2, and enables *module*-31 on the next rising clock edge instead of waiting for the overflow rippling in a standard ripple counter. The state look-ahead logic is principally equivalent to the one-cycle look-ahead mechanism in the counting path. The 4-bit counter state equation (counter state outputs of *module*-1 and *module*-31 in Fig. 5) can be expressed as:

$$Q3Q2Q1Q0 = Q3Q2Pipelined(q1q0) \tag{11}$$

where $Pipelined(X)$ denotes that the past bit values represented by X must be pipelined across one clock cycle. This notation may be recursively applied such that $Pipelined(Pipelined(X))$ would pipeline X across two clock cycles, and so forth. Consequently, the 6-bit counter state equation (counter state outputs of *module*-1, *module*-31, and *module*-32 in Fig. 5) can be expressed as:

$$Q5Q4Q3Q2Q1Q0Q5Q4 = Pipelined\big[(q3\overline{q2})Pipelined(q1q0)\big]. \tag{12}$$

The complete 8-bit counter state equation (counter state outputs of *module*-1, *module*-31, *module*-32, and *module*-33 in Fig. 5) can be expressed as:

$$\begin{aligned} &Q7Q6Q5Q4Q3Q2Q1Q0 \\ &\quad = Q7Q6Pipelined\big[(q5\overline{q4})Pipelined\big[(q3q2)Pipelined(\overline{q0q1})\big]\big] \end{aligned} \tag{13}$$

For (11), the past state *q1q0* is pipelined using the one-cycle look-ahead mechanism provided by the left-most module-2 in the counting path. For (12) and (13), the state look-ahead path provides the past states $\overline{q1}q0$ and $\overline{q1q0}$ through early overflow pipelining. Figure 8 depicts a generalized M-bit counter topology, revealing state look-ahead path details.

2.6 Counter's Logic Counts

In this subsection, we analyze the logic overhead of our parallel counter architecture based on the number of internal components (which can easily be translated to gate counts). *Module*-1 is a 2-bit counter that provides a set of early overflow states to enable future states. In general, if *module*-1 is a K-bit counter, one early overflow state inputs into the counting path and the remainder of the early overflow states input into the state look-ahead path.

The total number of Early Overflow States (*EOS*) generated by *module*-1 is

$$EOS = 2 * K - 1 \tag{14}$$

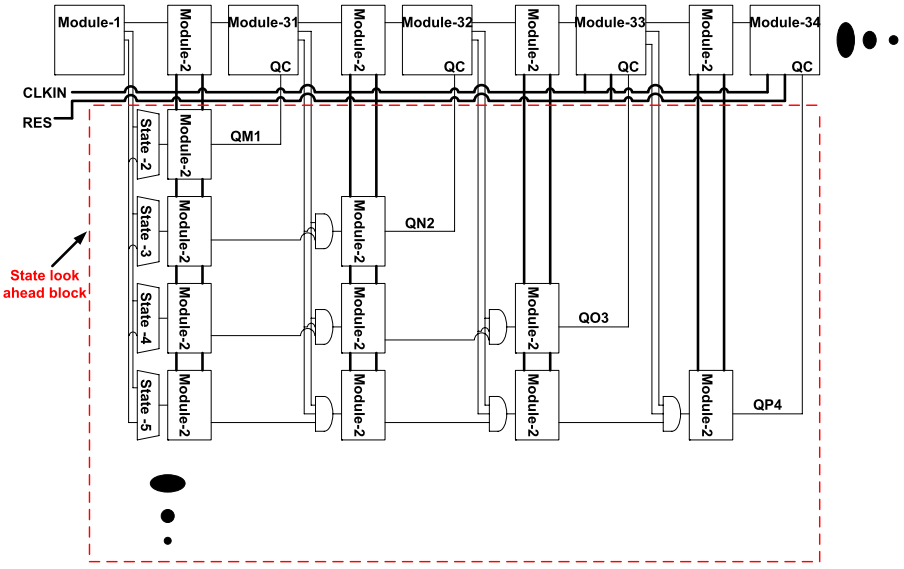

Fig. 8 Generalized counter topology for an M-bit counter showing state look-ahead path details

and the number of Early Overflow Components (EOC) (denoted as state-S in Fig. 8) required to propagate early overflow states in the state look-ahead path is

$$EOC = EOS - 1. \tag{15}$$

In order to generalize a skeleton structure to apply to variable counter widths, we consider the number of *module*-2*s* associated with each *module*-3S (one vertical column of *module*-2*s* for every *module*-3S in Fig. 8). One *module*-2 is in the counting path while the other *module*-2*s* are in the state look-ahead path. The number of Module-2*s* in the Vertical-column State Look-ahead Path ($M2_{VSLP}$) is

$$M2_{VSLP} = 2 * K - (S + 1), \tag{16}$$

where S denotes the horizontal position of *module*-3S.

The total number of *Module*-2s in each Vertical column ($M2_V$) (including both the counting and state look-ahead paths) is

$$M2_V = M2_{VSLP} + 1. \tag{17}$$

Using (14), the maximum allowable Counter Size (CS) in bits is

$$CS = K + \big(2(2 * K - 1)\big). \tag{18}$$

Using (18), the required number of *Module*-3*Ss* in the Counting Path ($M3_{CP}$), which is equal to the number of *Module*-2*s* in the Counting Path ($M2_{CP}$) is

$$M3_{CP} = M2_{CP} = (CS - K) * 2. \tag{19}$$

Table 1 Number of components and total counter range for sample counters with respect to the *module*-1 size in bits

Component	*Module-1 size*		
	Number of components		
	$K = 2$-bits	$K = 3$-bits	$K = 4$-bits
CS: Eq. (18)	8 (2^1 to 2^8)	17 (2^1 to 2^{17})	34 (2^1 to 2^{34})
$M3_{CP}$: Eq. (19)	3	7	15
*M*2: Eq. (21)	6	28	120
AND_{SLP}: Eq. (22)	1	15	91
EOC: Eq. (15)	2	6	14

The total number of *Module*-2*s* in the State Look-ahead Path ($M2_{SLP}$) is

$$M2_{SLP} = CS - 2 * K - CS - 2 * K - 1 + 12 \tag{20}$$

and the total number of *Module*-2*s* in the entire counter architecture ($M2$) is

$$M2 = M2_{CP} + M2_{SLP}. \tag{21}$$

The total number of AND logic in the State Look-ahead Path (AND_{SLP}) is

$$AND_{SLP} = CS - 2 * K - 2 * CS - 2 * K - 2 + 12. \tag{22}$$

Table 1 summarizes the component counts for various counter sizes based on the *module*-1 size in K bits.

2.7 Counter's Timing Delay

Using the proposed counter depicted in Fig. 5, we first derive the clock counter period T_{CLKIN} for an 8-bit counter based on the critical path signal propagation. Then, we generalize the derivation for an M-bit counter. The counter period must be greater than the worst-case critical path delay in the counting and the state look-ahead paths, such that

$$T_{CLKIN} > T_M + T_{AND3} + T_{SETUP\text{-}HOLD} \tag{23}$$

where T_M is the delay of *module*-3*S*, T_{AND3} is the delay of an 3-input AND-gate, and $T_{SETUP\text{-}HOLD}$ is the setup and hold time for the *module*-2. Furthermore, (23) can be represented as logic gate delays in order to avoid technology dependent factors, such that

$$\begin{aligned} &T_{SETUP\text{-}HOLD} = \frac{1}{2} * Inverter\_Delay; \qquad T_M = 2 * Inverter\_Delay; \\ &T_{AND3} = 1.5 * Inverter\_Delay \end{aligned} \tag{24}$$

resulting in

$$T_{CLKIN} = 4 * Inverter\_Delay. \tag{25}$$

Even though further speed enhancements are possible using various advanced design techniques for the *module*-1, *module*-2, and *module*-3*S*, the purpose of this paper is

to emphasize the architectural structure and parallel operation rather than improving individual components using high-cost technology and special circuit design techniques. These improvements are orthogonal to our work.

In order to derive the clock period for larger counter widths greater than 64-bit, the access time for the *module*-1 becomes the worst-case delay since larger *module*-1*s* generate more early overflow states as depicted in (14). The *module*-3*S* and *module*-2 components are independent of the counter size, and do not change in structure for larger bit widths (demonstrated in Fig. 8). *Module*-1*s* critical path delay is

$$T_{CLKIN} > T_{module\text{-}1} + T_{AND3} + T_{SETUP\text{-}HOLD}. \tag{26}$$

The key emphasis of (26) is that the delay is bounded by the *module*-1 for counter widths greater than 64-bits. In general, for very large counter widths (a *module*-1 size greater than 5 bits), further enhancements can be made using our topology as a sub-topology of a complete structure. Subsequently, the *module*-1 can be implemented using previous techniques [2, 7, 11, 26, 28, 30, 37, 52, 62], resulting in a critical path delay of

$$T_{CLKIN} > T_{module\text{-}1} - 1 * (T_M + T_{AND3} + T_{SETUP\text{-}HOLD}) + T_{AND3} + T_{SETUP\text{-}HOLD}. \tag{27}$$

T_{CLKIN} also has a secondary constraint due to the clock path's wire parasitic loading since the clock drives all modules in the structure simultaneously, which may be alleviated by using clock distributing and buffering approaches [18, 39] with an efficient layout implementation.

3 Simulation Results

To evaluate our proposed frequency divider, we present performance verification using HSPICE simulation results and power and area analysis. Figure 9 depicts the layout of our proposed frequency divider for a sample 8-bit divider implemented using

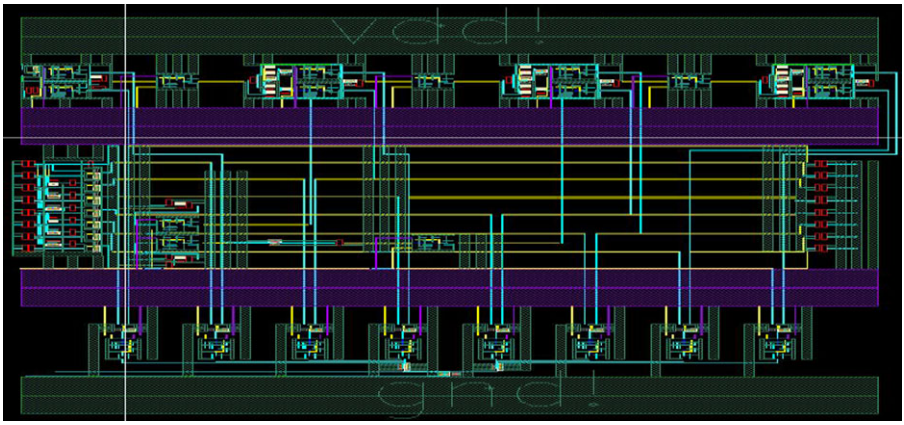

Fig. 9 Layout for our proposed frequency divider for a sample 8-bit divider

Berkley's Magic circuit layout tool [36]. We generated the HSPICE net-list from the Magic layout using resistance–capacitance parasitic extraction and performed performance verification using HSPICE 0.15-μm TSMC n-well CMOS technology [48] operating at 1.35 V and 125 °C, which provides worst-case corner delays [4].

3.1 Timing Analysis

Figure 10 depicts the HSPICE simulation waveform captured using the Mentor Graphics Powertrain waveform viewer [32]. The HSPICE simulation achieved a maximum operating frequency of 2 GHz for our 8-bit counter (Fig. 5) and the 8-bit divider (Fig. 2) with a safe margin of slew rate 0.1 ns/v rise/fall. The results show clean waveforms and fully functionality.

Figure 10 (a) depicts an exhaustive simulation of the parallel counter states obtained by toggling the input clock ($V(CLK)$) at 2 GHz frequency from ground-to-power supply and recording the output states. From top to bottom, Fig. 10(a) shows $V(q0) = V(CLK)/2$, $V(q1) = V(CLK)/4$, $V(q2) = V(CLK)/8$, $V(q3) = V(CLK)/16$, $V(q4) = V(CLK)/32$, $V(q5) = V(CLK)/64$, $V(q6) = (VCLK)/128$, and $V(q7) = V(CLK)/256$. The time scale on the horizontal axis is in nanoseconds and the vertical axis represents voltage with a maximum of 1.35 V. The states signals show sharp slew rates and clean waveforms indicating the design uses well driving buffers and a robust design circuitry to avoid glitches and unnecessary noise toggling even at very high input frequencies (e.g., 2 GHz in this example).

As an illustrative example, Fig. 10 (b) depicts the circuit simulations for the complete divider circuit for a dividing frequency $Fout = Fin/5$ by setting the programmable frequency signal to 5. From top to bottom, Fig. 10(b) shows $PQ0 = VDD$, $PQ1 = 0V$; $PQ2 = VDD$, $PQ3 = 0V$, $PQ4 = 0V$; $PQ5 = 0V$; $PQ6 = 0V$; and $PQ7 = 0V$. The input clock ($V(CLK)$) is running at 2 GHz, as is clearly shown with respect to the timing scale in nanoseconds on the horizontal axis. Figure 10(b) also shows the state signals $V(q0)$ and $V(q1)$ with respect to the pulse reload signal $V(RELP)$, which shows a period of 2.5 ns as is also clearly demonstrated by the delta measurement. These waveforms also reflect sufficient internal drivers' sizes with robust design and layout.

The maximum operating frequency can be theoretically verified using the derived (8) and (23). Combining these equations results in a total critical delay for our proposed frequency divider of

$$T_{CLKIN} \geq (T_M + T_{XOR} + T_{SETUP\text{-}HOLD}). \tag{28}$$

The delays of T_{AND3} and $T_{SETUP\text{-}HOLD}$ shown in (23) are not included in (28) since the path of the parallel counter that is used in setting the states for the next counter state modules is in parallel with the path of the parallel counter that is used in triggering the EOC, which is considered the critical path. In addition, (28) can be further presented as logic gate delays, in order to avoid technology-dependent factors,

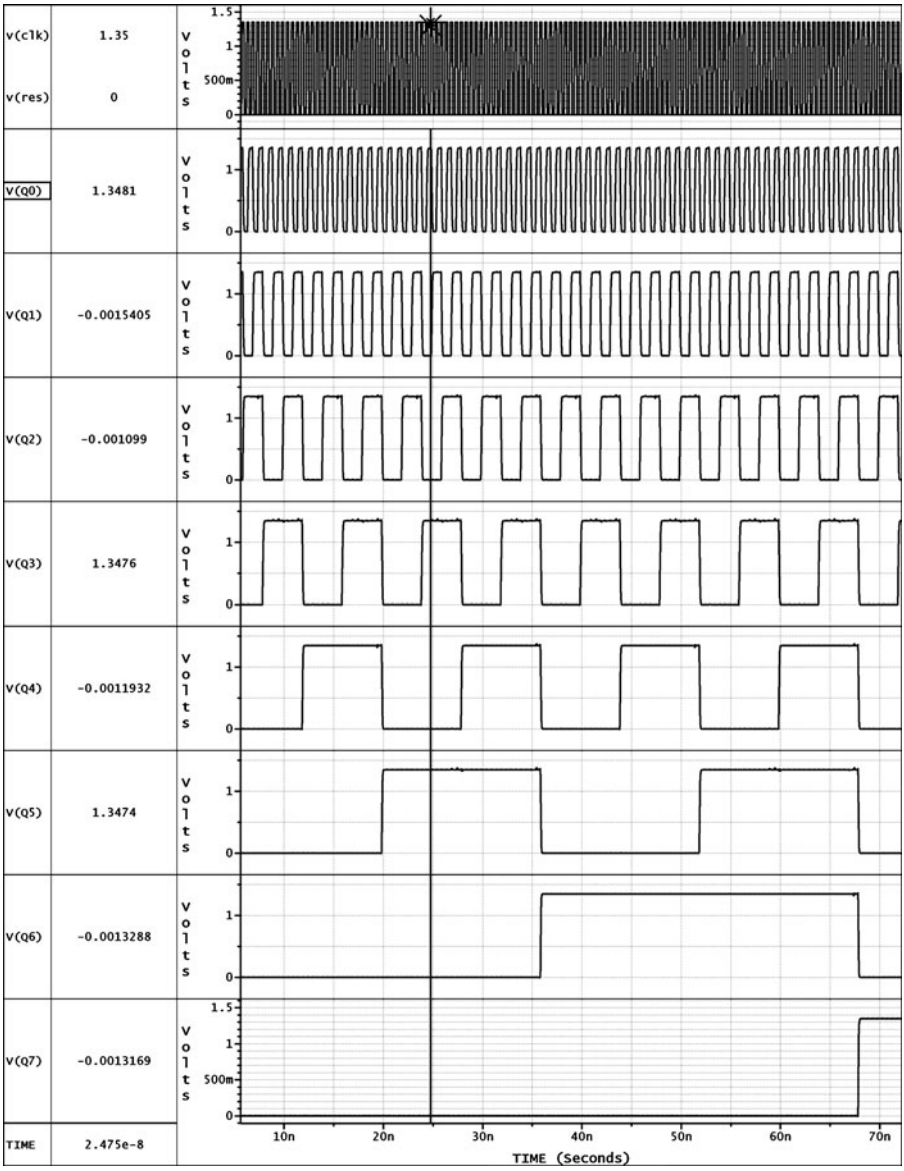

Fig. 10 (**a**) HSPICE waveforms for our 8-bit parallel counter ($CLK = 2$ GHz, $VDD = 1.35$ V) using 0.15-μm TSMC technology and (**b**) HSPICE waveforms for our 8-bit divider ($CLK = 2$ GHz, $VDD = 1.35$ V, $PQ0 = PQ2 = VDD$, $PQ1 = PQ3 = PQ4 = PQ5 = PQ6 = PQ7 = 0$) using 0.15-μm TSMC technology. The *horizontal scales* are in nanoseconds and the *vertical scales* are in volts

such that

$$T_M = 2 * Inverter\_Delay; \qquad T_{XOR} = 1.5 * Inverter\_Delay;$$
$$T_{SETUP\text{-}HOLD} = \frac{1}{2} * Inverter\_Delay \tag{29}$$

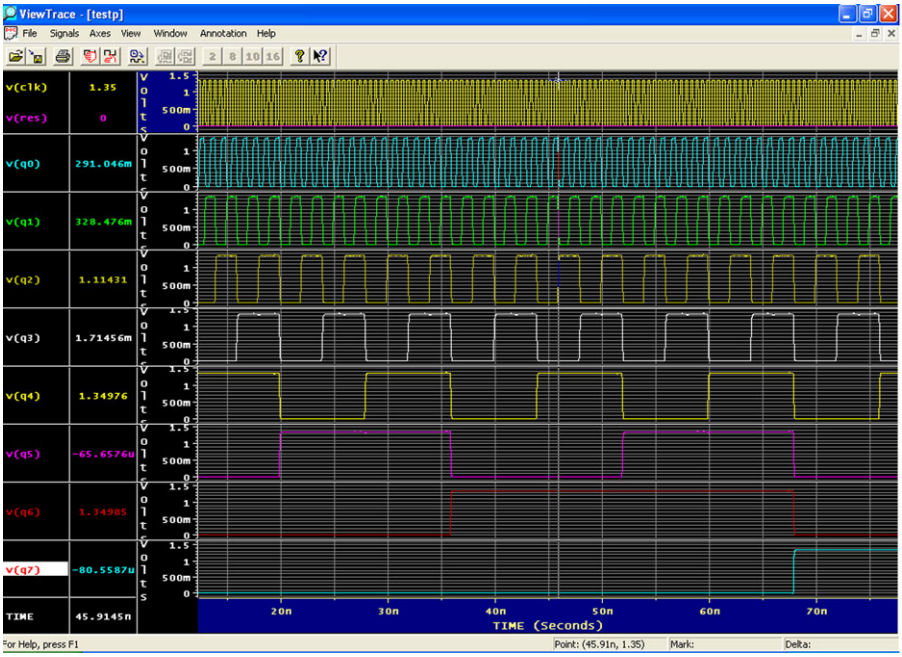

Fig. 10 (*Continued*)

resulting in

$$T_{CLKIN} \geq 4 * Inverter\_Delay. \tag{30}$$

On the other hand, we have depicted (9) as another alternative constraint on the operating frequency of the divider due to the chain of AND-tree logic in the reload circuit, which can impose a large delay constraint for large width dividers. However, this delay has a negligible effect since it increases in the form of $\mathrm{Log}_4(M)$ for an M-bit divider. For example, if the divider size is between 64 and 252 bits, the delay is only increased by one 4-input AND-gate logic. This delay of activating the reload time (T_{REL}) can be further reduced by inserting another pipeline of DFFs in the AND-tree logic of the reload circuit, thereby dividing this delay into two stages. Subsequently, the subtractor logic of the *EOC* needs to subtract a value of "3" instead of "2" in order to preserve zero pipeline latency frequency divider operation. Equation (9) can be further presented as logic gate delays such that

$$\begin{aligned} &T_{AND\text{-}TREE} = 2.5 * Inverter\_Delay; \qquad T_{PDFF\text{-}ACCESS} = 1 * Inverter\_Delay; \\ &T_{RESET\text{-}COUNTER} = 1.5 * Inverter\_Delay \end{aligned} \tag{31}$$

resulting in

$$T_{CLKIN} \geq 5 * Inverter\_Delay. \tag{32}$$

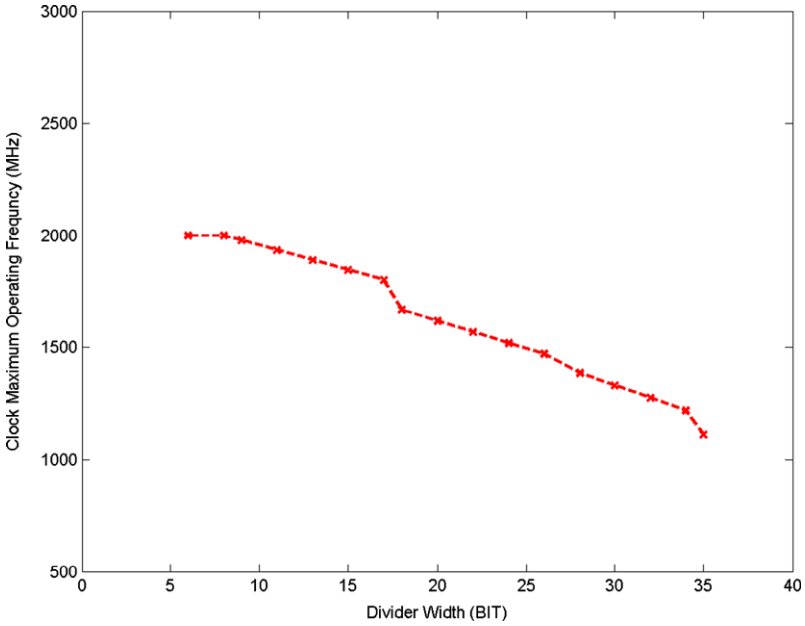

Fig. 11 Worst-case maximum clock frequency vs. divider width in bits

Equation (31) is the limiting factor in the operating frequency speed of our divider.

Figure 11 depicts worst-case maximum clock frequency vs. programmable divider bit-width. The simulation results show that the clock frequency decreases at an approximate rate of $\log_{6.5}(M)$ as the counter bit-width increases from 8 to 34. This decrease reflects the increasing cost of parasitic components on the system clock (*CLK*) due to an increase of *CLK*'s fan-out as the divider bit-width increases. The curve shows a sharp frequency decrease at 16 bits, which corresponds to the increased delay of the reload circuit, based on $[\mathrm{Log}_4(M)]$ due to the addition of one extra AND gate. A second sharp decrease occurs at 34 bits, which corresponds to the increase in the module-1 size to accommodate larger counter width as depicted in (14).

3.2 Power Analysis

Figure 12 depicts worst-case total power consumption (static and dynamic) vs. clock frequency for an 8-bit divider. We measure power consumption by setting the power supply voltage to 1.65 V at 0 °C for a worst-case process corner [4]. At clock frequencies of 2 GHz, 1 GHz, and 200 MHz, the divider consumes 15.479, 7.37, and 2.19 mW, respectively. Overall, power consumption increases at a moderate linear rate of 7.47 μW/MHz with respect to increasing clock frequency. The power consumption is primarily dominated by dynamic switching activity and local interconnects for all modules (which is at most a four-input-logic fan-in and a three-output-logic fan-out with the exception of *CLK* and *RELP*). The transistor size is optimized for high performance operation and low power by limiting the transistor width to

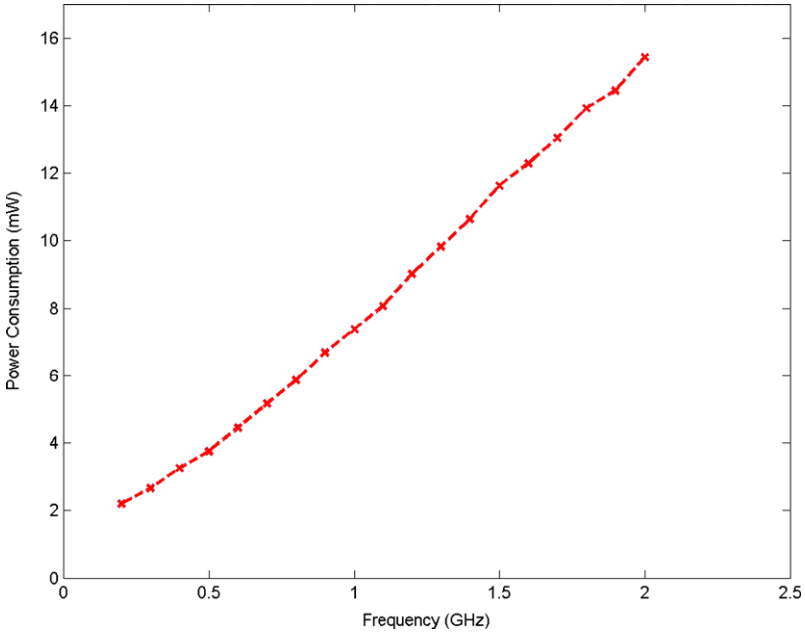

Fig. 12 Power consumption vs. clock frequency for an 8-bit divider

5 μm and the channel length to 0.15 μm, with the exception of the clock buffer transistors, which are approximately 15 μm in width and 0.15 μm in length, and thus minimizing the dynamic switching activity and preserving the static leakage power to the order of nano-watts. Overall, the power consumption increases at a moderate linear rate of 7.3 μW/MHz with respect to increasing clock frequencies. This small increase rate is due to the small geometry sizes of the modules since the modules are locally interconnected, which results in the elimination of global routing signals. Hence, the parasitic components are also minimized, which further reduces the power consumption. These factors are considered key factors for low-power CMOS design [5, 16, 48, 56].

3.3 Area Analysis

Table 2 summarizes the total number of components and transistors required per component for our 8-bit divider based on the component design modules depicted in Fig. 2 and Fig. 5 using CMOS transistor design structures. The main counter components, *module*-1, *module*-2, and *module*-3, require respectively 70, 90, and 24 transistors. The complete 8-bit divider requires only 900 transistors, which equals approximately 112,848 μm^2 of silicon die area. Figure 13 depicts the total transistor count vs. divider width in bits for divider ranges increasing by powers-of-two (transistors per range) and per 2-bit increments (transistors per 2-bits). Transistor requirements increase by approximately 3× for each power-of-two divider range increase (a subsequent increase in parallel counter modules). For divider widths ranging from 1 to 8 bits, 9 to 17 bits, 18 to 34 bits, and 35 to 67 bits, the parallel counter requires respectively 900,

Table 2 Total and per-component transistor requirements for an 8-bit divider

Component	Number of components	Transistors per component	Total transistor count
Module-1	1	70	70
Module-2	6	24	144
Module-3	3	90	270
States-S	2	9	18
3 input AND	1	8	8
EOC (XNOR)	8	12	96
EOC (SUB-1)	1	N/A	122
Reload	1	N/A	172
			Total = 900

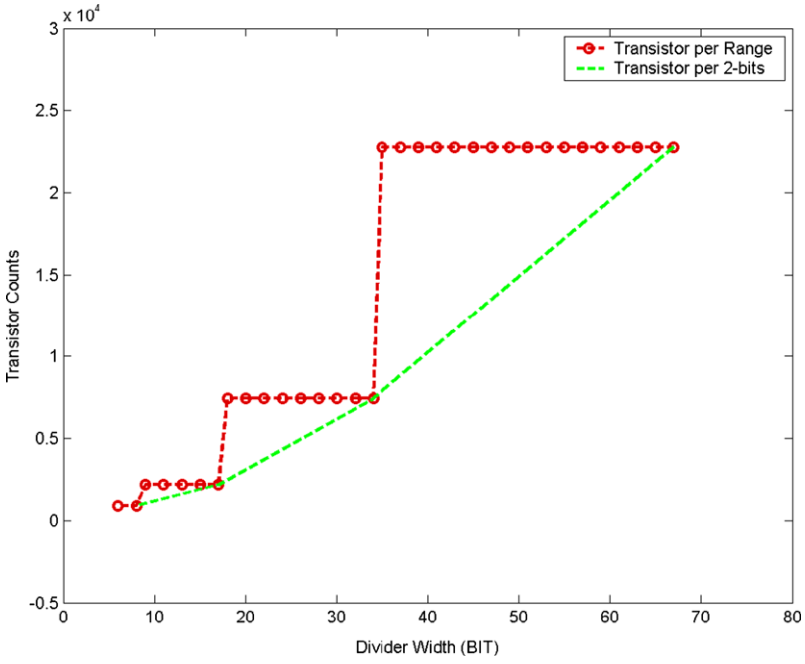

Fig. 13 Total CMOS transistor count versus divider width in bits

2,173, 7,426, and 22,784 transistors. Analyzed as the transistor increase per 2-bit increments in divider width reveals a modest linear increase of approximate 1.2× per 2-bits.

3.4 Comparison Results

Recent literature reveals many programmable divider designs with which to compare our proposed design; however, many factors complicate this comparison, such as different technology implementations, and more often, different technology simulation

Table 3 Comparison of our proposed frequency divider with previous work

	Tech.	Component design	Pipeline latency	Simulated max clock frequency (HSPICE)	Power at maximum frequency (HSPICE)	Area in number of digital CMOS transistors $N = 252$
Kuo and Wu [26]	0.18 µm	Not all digital: extended true-single-phase-clock for flip-flop design [41]	One clock cycle penalty at every counter restart	2 GHz (dividing factor $N = 33$)	8.48 mW (dividing factor $N = 33$)	576
Chang and Wu [7]	0.18 µm	All digital CMOS	One clock cycle penalty at every counter restart	1.305 GHz (dividing factor $N = 34$)	6.86 mW (dividing factor $N = 34$)	436
Proposed divider	0.15 µm	All digital CMOS	No latency due to subtractor circuit	2 GHz (dividing factor $N = 252$)	15.47 mW (dividing factor $N = 252$)	900

environments (even though the designs might have the same aligned technology). Another important aspect is the cell implementation of the design, where many modern implementations target their design using dynamic and analog concepts of cells design in order to operate for high-range frequencies with optimized power consumption.

Since our architecture is a divide-by-N frequency divider and is implemented entirely with digital CMOS circuits, we compare our results to the ones published by Kuo and Wu [26] and Chang and Wu [7] in order to have a close comparison.

Table 3 summarizes the comparison in terms of power consumption in milliwatts, maximum clock frequency in gigahertz, and area requirements in number of transistors for an 8-bit divider (maximum dividing factor $N = 252$). The power consumption of [26] measured at 2 GHz is the least power consumption of the compared designs due to the use of extended true-single-phase-clock for flip-flop components [13]. Their controller circuit (analogous to our proposed architecture's *Reload* and *EOC*) is only activated during the pulse of the selected dividing factor and requires a pulse generator circuit. Their circuit provides operating speeds up to 2 GHz at the cost of adding one extra cycle (pipeline latency) for every dividing frequency and using high-cost circuit cell design techniques. Reference [7] trades maximum operating frequency for low power consumption (6.86 mW) due to small transistor counts and low operating frequency. Our proposed design's transistor count is larger due to the parallel counter with the state look-ahead mechanism. The modules in our proposed divider are all locally interconnected regardless of the divider width, which inhibits a small transistor size. Subsequently, the dynamic power of our design is reduced, which also can be further reduced by using advance circuit techniques [3, 13, 15, 22, 35, 43, 49, 51, 58]. On the other hand, as the frequency divider value N increases, our divider's transistor count increases at a rate of 1.2 per 4-bit increase, as is depicted in Fig. 13. Thus, our divider's power dissipation is still larger compared

to previous work even for very large N values. Nevertheless, our divider reaches operating speeds of 2 GHz with a maximum full range of exhaustive dividing frequencies ($N = 256$) without any additional clock cycle penalty usually introduced by a pipeline structure. Finally, our divider leverages the low cost of basic digital CMOS components with a scalable structure that is independent of additional gate delay penalties to the operating frequency.

4 Conclusions

In this paper, we present a modular structured high-speed wide-range divide-by-N frequency divider. The divider structure's main features use pipelining and state look-ahead logic whose interoperation activates all modules concurrently on the system's clock edge. The divider's critical path is divided into three cycles: the first cycle for state evaluation and end-of-count (EOC) detection, the second cycle for reload pulse activation and resetting the parallel counter, and the third cycle for deactivating the reload circuit and releasing the counter to restart counting on the next clock edge. Subtractor arithmetic in conjunction with the end-of-count detector (EOC) is used to subtract a value of "2" from the current count value in order to eliminate all pipeline latency cycles, resulting in exact frequency division. A divider of size M bits has a range from 3 (instead of 2) to 2^M due to the subtractor and provides the zero pipeline latency with a negligible range loss. Results reveal that our divider frequency drops at a rate of $\log_{6.5}(M)$ (where M is the divider width in bits) due to parasitic components that inhibit large fan-out of the system clock path (since all modules operate simultaneously). This logarithmic frequency drop places our design amongst the fastest wide-range divider designs reported in literature, to the best of our knowledge. Furthermore, area requirements increase at a modest rate of $1.2\times$ transistors per 2-bit increase in divider width. Similarly, power consumption increases at a moderate linear rate of 7.47 μW/MHz with respect to each doubling of the clock frequency (two-bit increase). Finally, the divider output value is determined directly, on-the-fly with no additional decoding latency necessary to decode the final output results.

Acknowledgements This work was supported in part by the National Science Foundation (CNS-0953447). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

1. S. Abdel-Hafeez, S.M. Harb, W.R. Eisenstadt, High speed digital CMOS divide-by-N frequency divider, in *IEEE International Symposium on Circuits and Systems (ISCAS 2008)*, May 2008 (2008), pp. 592–595
2. M. Ali, E. Hegazi, A multigigahertz multimodulus frequency divider in 90-nm CMOS. IEEE Trans. Circuits Syst. II, Express Briefs **53**, 1333–1337 (2006)
3. S.-Y. Bai, T.-N. Luo, Y.-J.E. Chen, A 1 V 50 GHz digital-controlled CMOS frequency divider, in *Asia Pacific Microwave Conference (APMC 2006)*, Dec. 2006 (2006), pp. 755–758
4. A. Bellaour, M.I. Elmasry, *Low-Power Digital VLSI Design Circuits and Systems* (Kluwer Academic, Norwell, 1995)

5. W. Belluomini, D. Jamsek, A.K. Martin, C. McDowell, R.K. Montoye, H.C. Ngo, J. Sawada, Limited switch dynamic logic circuits for high-speed low-power circuit design. IBM J. Res. Dev. **50**(2/3), 277–286 (2006)
6. J. Chabloz, D. Ruffieux, C. Enz, A low-power programmable dynamic frequency divider, in *34th European Solid-State Circuits Conference (ESSCIRC 2008)*, Sept. 2008 (2008), pp. 370–373
7. H. Chang, J. Wu, A 723-MHz 17.2-mW CMOS programmable counter. IEEE J. Solid-State Circuits **33**, 1572–1575 (1998)
8. K. Cheng, C. Hung, C. Su, A sub-1V low-power high-speed static frequency divider, in *IEEE International Symposium on Circuits and Systems (ISCAS 2007)*, May 2007 (2007), pp. 3848–3851
9. J. Chien, L. Lu, Ultra-low-voltage CMOS static frequency divider, in *Asian Solid-State Circuits Conference*, Nov. 2005 (2005), pp. 209–212
10. M. Chuang, J. Kuo, C. Wang, H. Wang, A 50 GHz divide-by-4 injection lock frequency divider using matching method. IEEE Microw. Wirel. Compon. Lett. **18**(5), 344–346 (2008)
11. H.-I. Cong, J.M. Andrews, D.M. Boulin, S.-C. Fang, S.J. Hillenius, J.A. Michejda, Multigigahertz CMOS dual-modulus prescalar IC. IEEE J. Solid-State Circuits **23**, 1189–1194 (1988)
12. P. Coussy, A. Morawiec, *High-Level Synthesis: From Algorithm to Digital Circuit* (Springer, Berlin, 2008)
13. M.A. Do, X.P. Yu, J.G. Ma, K.S. Yeo, R. Wu, Q.X. Zhang, A 2 GHz programmable counter with new reloadable D flip-flop, in *IEEE Conference on Electron Devices and Solid-State Circuits*, Dec. 2003 (2003), pp. 269–272
14. M.A. Do, X.P. Yu, J.G. Ma, K.S. Yeo, R. Wu, Q.X. Zhang, GHz programmable counter with low power consumption. IEEE Electron. Lett. **39**, 1572–1573 (2003)
15. A. Fard, D. Aberg, A novel 18 GHz 1.3 mW CMOS frequency divider with high input sensitivity, in *International Symposium on Signals, Circuits and Systems (ISSCS 2005)*, July 2005, vol. 2 (2005), pp. 409–412
16. N. Foroudi, S. Fulga, P. Suppiah, J.N.M. Peirce, Low-voltage low-power topology for high-speed applications, in *Proceedings of the Bipolar/BiCMOS Circuits and Technology Meeting*, Oct. 2001 (2001), pp. 135–138
17. Z. Gu, A. Thiede, 18 GHz low-power CMOS static frequency divider. Electron. Lett. **39**, 1433–1434 (2003)
18. A. Hastings, *Art of Analog Layout* (Prentice Hall, New York, 2005)
19. C. He, Y.H. Sun, Design of 10 GHz, 2.6 mW frequency divider in 0.25 μm CMOS, in *6th International Conference on Solid-State and Integrated-Circuit Technology*, Oct. 2001, vol. 2 (2001), pp. 1136–1138
20. S. Henzler, S. Koeppe, High-speed low-power frequency divider with intrinsic phase rotator, in *International Symposium on Low Power Electronics and Design (ISLPED '06)*, Oct. 2006 (2006), pp. 286–291
21. S. Henzler, S. Koeppe, Design and application of power optimized high-speed CMOS frequency dividers, in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Nov. 2008, vol. 16 (2008), pp. 1513–1520
22. P. Heyari, R. Mohhanavelu, A 40-GHz flip-flop-based frequency divider. IEEE Trans. Circuits Syst. II, Express Briefs **53**, 1358–1362 (2006)
23. Q. Hu, H.-A. Zhao, C. Liu, A programmable frequency divider in 0.18 μm CMOS library, in *IEEE Computer Society Annual Symposium on VLSI (ISVLSI '08)*, April 2008 (2008), pp. 157–161
24. Y. Kado, M. Suzuki, Y. Okazaki, T. Kobayashi, 1 GHz CMOS 1/8 static frequency divider operating at 1 V. Electron. Lett. **28**(16), 167–169 (1992)
25. M.G. Kane, P.Y. Chan, S.S. Cherensky, D.C. Fowlis, A 1.5-GHz programmable divide-by-*N* GaAs counter. IEEE J. Solid-State Circuits **23**, 480–484 (1988)
26. K.-C. Kuo, F.-J. Wu, A 2.4-GHz/5-GHz low power pulse swallow counter in 0.18-μm CMOS technology, in *IEEE Asia Pacific Conference on Circuits and Systems (APCCAS 2006)*, Dec. 2006 (2006), pp. 214–217
27. B.L. Liang, D. Chen, B. Wang, D. Cheng, T. Kwasniewski, A 43-GHz static frequency divider in 0.13 μm standard CMOS, in *Canadian Conference on Electrical and Computer Engineering (CCECE 2008)*, May 2008 (2008), pp. 111–114
28. P. Larsson, A wide-range programmable high-speed CMOS frequency divider, in *IEEE International Symposium on Circuits and Systems (ISCAS '95)*, May 1995, vol. 1 (1995), pp. 195–198
29. P. Larsson, High-speed architecture for a programmable frequency divider and a dual-modulus prescaler. IEEE J. Solid-State Circuits **31**, 744–748 (1996)

30. S. Lee, H.J. Park, A CMOS high-speed wide-range programmable counter. IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process. **49**, 638–642 (2002)
31. D. Mandal, T.K. Bhattacharyya, Implementation of CMOS low-power integer-N frequency synthesizer for SOC design. J. Comput. **3**(4), 31–38 (2008)
32. Mentor Graphics, Wilsonville, Oregon, 1981. http://www.mentor.com/
33. F.P.H. de Miranda, S.J. Navarro Jr., W.A.M. Van Noije, A 4 GHz dual modulus divider-by 32/33 prescaler in 0.35 µm CMOS technology, in *17th Symposium on Integrated Circuits and Systems Design (SBCCI 2004)*, Sept. 2004 (2004), pp. 94–99
34. M. Nogawa, Y. Ohtomo, A 16.3-GHz 64:1 CMOS frequency divider, in *Conference on Proceedings of the Second IEEE ASIA Pacific (ASICs 2000)*, Aug. 2000 (2000), pp. 95–98
35. R. Nonis, E. Palumbo, P. Palestri, L. Selmi, A design methodology for CMOS current-mode logic frequency dividers. IEEE Trans. Circuits. Syst. I, Regul. Pap. **54**, 245–254 (2007)
36. J. Ousterhout, Open circuit design. Berkeley (1980). http://opencircuitdesign.com/magic/
37. S. Pellerano, S. Levantino, C. Samori, A.L. Lacaita, A 13.5-mW 5-GHz frequency synthesizer with dynamic-logic frequency divider. IEEE J. Solid-State Circuits **39**, 378–383 (2004)
38. J.O. Plouchart, J. Kim, H. Recoules, N. Zamdmer, Y. Tan, M. Sherony, A. Ray, L. Wagner, A 0.123 mW 7.25 GHz static frequency divider by 8 in a 120-nm SOI technology, in *International Symposium on Proceedings of the Low Power Electronics and Design (ISLPED '03)*, Aug. 2003 (2003), pp. 440–442
39. B. Razavi, *Fundamentals of Microelectronics* (Wiley, New York, 2008)
40. B. Razavi, K.F. Lee, R.H. Yan, Design of high-speed, low-power frequency dividers and phase-locked loops in deep submicron CMOS. IEEE J. Solid-State Circuits **30**(2), 101–109 (1995)
41. A. Safarian, S. Anand, P. Heydari, On the dynamics of regenerative frequency dividers. IEEE Trans. Circuits Syst. II, Express Briefs **53**(12), 1413–1417 (2006)
42. K. Sengupta, H. Hashemi, Maximum frequency of operation of CMOS static frequency dividers: theory and design techniques, in *13th IEEE International Conference on Electronics, Circuits AND Systems 2006 (ICECS '2006)*, Dec. 2006 (2006), pp. 584–587
43. K.M. Sharaf, A low-power CMOS multi-modulus dynamic frequency divider, in *National Radio Science Conference*, March 2008 (2008), pp. 1–9
44. K. Shu, E. Snchez-Sinencio, *CMOS PLL Synthesizers: Analysis and Design* (Springer, Berlin, 2004)
45. U. Singh, M. Green, New structures for very high-frequency CMOS clock dividers, in *IEEE International Symposium on Circuits and Systems (ISCAS 2001)*, May 2001, vol. 4 (2001), pp. 622–525
46. S.B. Sleiman, J.G. Atallah, S. Rodriguez, A. Rusu, M. Ismail, Wide-division-range high-speed fully programmable frequency divider, in *6th International IEEE Northeast Workshop on Circuits and Systems and Join TAISA Conference (NEWCAS-TAISA 2008)*, June 2008 (2008), pp. 17–20
47. J.N. Soares, W.A.M. Van Noije, A 1.6-GHz dual modulus prescaler using the extended true-single-phase-clock CMOS circuit technique (E-TSPC). IEEE J. Solid-State Circuits **34**, 97–102 (1999)
48. Taiwan Semiconductor Manufacturing Company Ltd, Taiwan. http://www.tsmc.com
49. S. Tseng, C. Meng, S. Li, J. Su, G. Huang, 2.4 GHz divide-by-256/271 single-ended frequency divider in standard 0.35-µm CMOS technology, in *Microwave Conference Proceedings and Asia_Pacific Conference Proceedings (APMC 2005)*, Dec. 2005, vol. 2 (2005), pp. 4–6
50. P. Upadhyaya, G.S. La Rue, Static frequency divider with enhanced frequency performance, in *IEEE Workshop on Microelectronics and Electron Devices (WMED 2008)*, April 2008 (2008), pp. 20–21
51. M. Usama, T.A. Kwasniewski, A 40-GHz frequency in 90-nm CMOS technology, in *IEEE North-East Workshop on Circuits and Systems*, June 2006 (2006), pp. 41–43
52. C.S. Vaucher, I. Ferencic, M. Locher, S. Sedvallson, U. Voegeli, Z. Wang, A family of low-power truly modular programmable dividers in standard 0.35-µm CMOS technology. IEEE J. Solid-State Circuits **35**, 1039–1045 (2000)
53. H. Wang, A 1.8 V 3 mW 16.8 GHz frequency divider in 0.25 µm CMOS, in *IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC 2000)*, Feb. 2000 (2000), pp. 196–197
54. H. Wang, M.L. Yongming, Y. Li, H. Chen, A novel high-speed programmable counter architecture for 5 GHz WLAN application, in *International Symposium on Signals, Circuits, and Systems (SCS 2003)*, July 2003, vol. 1 (2003), pp. 233–236
55. J.M.C. Wong, V.S.L. Cheung, H.C. Luong, A 1-V 2.5-mW 5.2-GHz frequency divider in a 0.35-µm CMOS process. IEEE J. Solid-State Circuits **38**, 1643–1648 (2003)
56. A. Wroblewski, C.V. Schimpfle, O. Schumacher, J.A. Nossek, Minimizing spurious switching activities with transistor sizing, in *VLSI Design*, vol. 15 (Taylor & Francis, London, 2002), pp. 537–545

57. K. Yamamoto, M. Fujishima, 4.3 GHz 44 μW CMOS frequency divider, in *IEEE International Solid-State Circuits Conference, Digest of Technical Papers (ISSCC 2004)*, Feb. 2004, vol. 1 (2004), pp. 104–116
58. W. Yang, J. Cao, F. Ran, T. Qing, A high-speed divide-by-32/33 frequency divider in 0.25 μm CMOS technology, in *4th International Conference on Microwave and Millimeter Wave Technology (ICMMT 2004)*, Aug. 2004 (2004), pp. 554–557
59. C. Yang, G. Dehng, J. Hsu, S. Liu, New dynamic flip-flops for high-speed dual modulus prescaler. IEEE J. Solid-State Circuits **33**, 1568–1571 (1998)
60. X.P. Yu, M.A. Do, L. Jia, J.G. Ma, K.S. Yeo, Design of a low power wide-band high resolution programmable frequency divider, in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Sept. 2005, vol. 13 (2005), pp. 1098–1103
61. X.P. Yu, W.M. Lim, M.A. Do, X.L. Yan, K.S. Yeo, 6.1 GHz 4.6 mW CMOS divide-by-55/56 prescaler. Electron. Lett. **44**(20), 1402–1403 (2008)
62. M. Zhang, S.K. Islam, M.R. Haider, A fast programmable frequency divider with a wide dividing-ratio range and 50% duty-cycle. IEICE Electron. Express **4**(21), 672–678 (2007)