Performance Enhancement Counter with Minimal Clock Period

Anto Mamman Shane S¹, Muralidharan J²

¹PG Scholar, Department of Electronics and Communication Engineering, KPR Institute of Engineering and Technology, Coimbatore, Tamilnadu, India
²Associate Professor, Department of Electronics and Communication Engineering, KPR Institute of Engineering and Technology, Coimbatore, Tamilnadu, India

Abstract. A synchronous binary counter is a fundamental component in VLSI design which are used commonly. synchronous binary counter is fast and are used in many applications as it supports wide bit-width. Due to large fan-outs and long carry chains many previous counters have low counting rate when the size of the counters is large. A new fast structure has been suggested for synchronous binary counter with a very low delay for counter with size ranging from 8 to 128 bits. To reduce the complexity of hardware a 1-bit Johnson counter has been used and then duplicate it to minimise propagation delay induced by large fan-outs. The suggested design is realised with a small number of flip-flops, using a back carry propagation counter and a counter based on state look ahead logic, which reduces power and delay.

1 Introduction

The utilization of a 1-bit Johnson counter decreases hardware complexity and is replicated to minimize the delay caused by high fan-out connections. This design strategy employs a limited number of flip-flops and includes a back carry propagation counter and a counter that relies on state look-ahead logic. This helps reduce power consumption and delay. Asynchronous counters, also known as ripple counters, are implemented with logic gates and have fewer gates compared to synchronous counters. However, they suffer from accumulated delay due to ripple propagation, leading to occasional false outputs when flip-flops are not connected to the same clock signal. This ripple effect becomes more pronounced in larger counter sizes, which can be problematic in applications where stable counting values are required. Synchronous binary counters, such as ripple carry counters, use a carry chain where the carry-out is connected to the carry-in of the next step. The propagation delay caused by the carry chain limits the speed of synchronous counters. Various techniques have been developed to improve the speed of adders, which are also employed to enhance counter performance. To achieve significant speed improvements, carry-look ahead circuits are used instead of ripple carry chains in traditional binary counters. Hierarchical Manchester carry chains and state-look ahead topologies have been employed to address carry propagation issues. Carry chains have been built using a tree framework, and attempts have been made to increase counter speed by using better flip-flops. Synchronous counters with high speed have been developed by utilizing flip-flops based on true single-phase clocks.
Analog-to-digital converters (ADCs) require a synchronous counter that can operate efficiently over a broad frequency range. Aside from its importance in ADCs, it has been discovered to be the fundamental building component for a variety of usages in the fields such as VLSI design, communication systems, and many measuring systems. As a result, significant research is being conducted by researchers on to design counters with rigorous speed and efficient power consumption. Synchronous counters with small frequencies to counters with average frequencies applications are mainly built by consecutive stages, with every stage having combinational logic and a T flip-flop or D flip-flop. Even though common clock triggers every TFFs at the same time the choice to toggle is driven by the combinational logic. A carry propagation chain with inputs from previous levels outputs are typical combinational logic. Nonetheless, we are interested in fast counters for two purposes. Counters are essential tools for debugging and measuring digital systems, and they need to be faster than the system being observed. Therefore, it is crucial to design counters that are optimized for the given technologies to achieve maximum speed. Binary arithmetic circuit theory has made significant progress by achieving an initial state with logarithmic time complexity for n-bit addition and multiplication logic. However, this theory, which is applicable to combinational circuits used for instrumentation, fails to provide the same efficiency for synchronous counters [2-8]. Researchers conducted a study that showcased this constraint by creating a synchronous binary counter with a clock period unaffected by the counter's size. The study revealed that addition operations were significantly slower compared to instrumentation tasks. The counter utilized in this context is comparable to, but slightly simpler than, a counterpart with the same clock speed. In a 64-bit comparison test version, our counter requires nearly twice the area compared to the counterpart mentioned in the study.

Counters have a vital function in digital systems, but achieving both high speed and a large counter size simultaneously can be difficult because of the carry propagation that occurs from low-order bits to high-order bits. Designing counters that are both long and fast requires the utilization of rescaling techniques. In the industry, there are examples of counters that achieve this combination of length and speed. These counters exhibit a clock interval that remains constant, regardless of the counter's size, thanks to the use of rescaling techniques. According to complexity theory, a pre-scaled counter has a constant clock interval, represented as O(1), regardless of the counter's size N. However, in practice, this ideal scenario is only partially true, as it assumes a synchronous paradigm with a perfect broadcast clock signal.

In synchronous designs, the sampling rate is equal to the counting rate, as they operate based on a clock signal. This property is advantageous for achieving precise timing and synchronization. However, in cases where synchronous behavior is not necessary, a simplified ripple-carry (asynchronous) counter can be used, especially in applications such as frequency dividers. The complexity of space in counters is typically proportional to the number of bits, denoted as O(N). As the counter's bit count grows, the complexity of the design and the necessary resources also increase. For counters with larger asymptotic complexity, ring counters can be used, which have a space complexity of O(2N). Linear Feedback Shift Registers (LFSRs) are commonly used when binary sequences are not required. LFSRs can have periods ranging from O(1) to O(N) space complexity, depending on the specific design. Counters can be categorized as "up," "down," or "up-down" counters. Up-only counters produce output sequences that increase and wrap around modulo-2N, while down-only counters produce decreasing output sequences modulo-2N. Up-down counters have the ability to change their counting direction in any clock cycle when an input signal is applied. They can also be loaded with a new initial value, allowing them to count from the new value in subsequent cycles. Achieving counters with constant time and this property can be challenging and is not specifically addressed in the mentioned study. For certain applications, such as maximizing data ingest rate, Rapid Single Flux Quantum (RSFQ)
counters are suitable. These counters are designed using RSFQ technology, which offers advantages in terms of circuit design, fabrication, and hardware testing. They are commonly used as digital benchmark circuits.

2 Literature Survey

A novel and efficient design for a fast synchronous binary counter with a short delay interval is introduced [9]. This design can handle counter sizes ranging from 8 to 128 bits. It stands out by employing a minimal number of flip-flops, approximately equal to the counter size, and operates at a low clock frequency. To achieve high operating frequencies, a wide-range parallel counter [10] is combined with a standard pipeline partitioning technique. The counter structure includes three replicated CMOS-logic modules comprising a state look-ahead path, a simple D-type flip-flop, and a 2-bit counter. The state look-ahead path enables simultaneous activation of all modules at the clock edge, ensuring consistent delay in updating count states across the counting path modules/stages. The paper also explores the use of LFSR (Linear Feedback Shift Register) counters as an alternative approach for large counter arrays, offering improved area efficiency and performance compared to binary counters. However, decoding the count order into binary requires additional complex logic, making it impractical for system-on-chip designs. The paper presents a counter design based on multiple LFSRs that retains the benefits of a single-stage LFSR while minimizing the need for decoding logic.

A heart rate sensor is utilized to measure pulse waves, which indicate variations in blood vessel volume during blood circulation. These pulse waves are detected by an optical sensor and a green LED, enabling the identification of changes in volume. The typical heart rate range is between 60 and 100 beats per minute, and the application operates at a frequency of 800 MHz [11]. To facilitate the summation of seven inputs, a 7-to-3 parallel counter was developed and tested using a dedicated test bed. A novel and efficient design for a fast synchronous binary counter with a short delay interval is introduced. This design can handle counter sizes ranging from 8 to 128 bits. It stands out by employing a minimal number of flip-flops, approximately equal to the counter size, and operates at a low clock frequency. To achieve high operating frequencies, a wide-range parallel counter [10] is combined with a standard pipeline partitioning technique. The counter structure includes three replicated CMOS-logic modules comprising a state look-ahead path, a simple D-type flip-flop, and a 2-bit counter. The state look-ahead path enables simultaneous activation of all modules at the clock edge, ensuring consistent delay in updating count states across the counting path modules/stages. The paper also explores the use of LFSR (Linear Feedback Shift Register) counters as an alternative approach for large counter arrays, offering improved area efficiency and performance compared to binary counters. However, decoding the count order into binary requires additional complex logic, making it impractical for system-on-chip designs. The paper presents a counter design based on multiple LFSRs that retains the benefits of a single-stage LFSR while minimizing the need for decoding logic.

3 Proposed Methodology

Figure 1 illustrates a traditional synchronous binary counter composed of T flip-flops (F/Fs) with XOR gates used to manipulate bit values. XOR gates are the simplest logic elements employed in this counter design. To simplify the design, an XOR gate and a D flip-flop are combined into a T flip-flop. The lower bits of the counter are sequentially examined using AND gates, starting from the Least Significant Bit (LSB). If all lower bits are set to 1, an AND gate produces a carry signal. These AND gates are connected in a serial manner, forming a ripple carry chain. The ripple carry chain transfers the carry generated by the LSB
to the Most Significant Bit (MSB). However, the carry output of an AND gate becomes invalid if any lower carry signals are unstable. The delay in the propagation path of the counter does not grow exponentially as the counter size increases, resulting in a significant delay proportional to the counter size. This delay is responsible for the characteristic delay observed in a conventional binary counter.

Fig. 1. 8-bit synchronous binary counter with a ripple carry chain (Conventional).

The concept of backward carry propagation, introduced in [12], is crucial for achieving a fast synchronous counter. In a binary number system, the most significant bit of the counter undergoes a state change earlier than the least significant bit. Instead of using a single chain like in conventional binary counters, each bit of the counter has its own dedicated AND chain, connected in reverse order. This arrangement allows the signals in the carry chain to be evaluated in the order they arrive, prioritizing the earlier signals. Signals that change rapidly from the least significant bit are connected to the last AND gate in the carry chain. The critical path delay in the backward carry propagation is determined by the delay of this final AND gate and the T flip-flop. However, the least significant bit (LSB) is connected to every AND chain, resulting in large fan-outs. Large fan-outs can cause delays in driving the load of the LSB, which subsequently affects the critical path that depends on the fan-out.

Fig. 2. 8-bit synchronous binary counter that employs backward carry propagation.

The constant-delay binary counter, illustrated in Figure 3, divides a wide counter design into smaller sub-blocks, with the high-order sub-block operating at a lower frequency than the low-order sub-block. A pre-scaler block is utilized in the low-order sub-block to generate a pre-scaling enable (PEN) signal, which has a much lower frequency than the clock signal. This enables the high-order block to be incremented less frequently compared to the low-order block. Generating the PEN signal involves using a ring counter, which introduces complexity to the design. The complexity of the counter is primarily influenced by the least significant block, comprising an XOR gate, a D flip-flop's loading delay, and a D flip-flop's setup time. The count enable signal (CNT) is distributed through fan-out, while the PEN signal significantly reduces the counting rate. However, as the counter size increases, the issue of fan-out becomes more problematic, leading to an escalation in the propagation delay.
of the PEN signal. The actual delay is influenced by nodes with large fan-outs. In the proposed approach, the complexity and delay caused by fan-outs are mitigated by employing replicated 1-bit Johnson counters and implementing backward carry propagation to generate PEN signals. This enables the proposed counter to achieve a higher count rate for practical counter sizes.

![Fig. 3. 64-bit prescaled counter](image)

The N-bit counter, depicted in Figure 4, employs pre-scaling using a straightforward logic based on \( n = \log_2 N \) and \( m = (N - n)/L \), where \( L \) represents the large fan-outs determined through simulation. It is divided into three sub-counters, with \( m \) 1-bit Johnson counters generating PEN signals for the final sub-counter. When a Johnson counter changes state from 0 to 1, a PEN signal is produced to enable counting in the next sub-counter. Each counter block functions as a sequence generator, incrementing the count from 00...000 to 11...111. In general, a counter consists of registers for storing the current state and an incrementor for calculating the next value.

![Fig. 4. Detailed structure representation of its components and organization of the proposed N-bit counter.](image)

The anticipated architecture for the N-bit counter, as shown in Figure 4, divides the counter into three sub-counters: C1, C2, and C3. Each sub-counter serves a specific purpose within the overall design. C1 acts as a 1-bit counter that alternates between the values 1 and 0. C2, on the other hand, operates as an \((n-1)\)-bit counter using backward carry transmission. Finally, C3 is responsible for handling the remaining \((N-n)\) bits and operates as a state look-ahead logic binary counter. The key concept behind this architecture is the use of pre-scaling, which allows for separate treatment of the high-order and low-order blocks of the counter. This approach enables more efficient and optimized processing. In particular, the gate delay of C3, which includes \((N-n-1)\) AND gates and operates with synchronous ripple carry, is significantly shorter than the duration of the PEN2 signal generated by C2. PEN2 has a
duration of $2^n$ clock cycles, ensuring that carry propagation within C3 stabilizes before the next PEN2 signal from C2 is received. By utilizing backward carry propagation, C2 minimizes the delay caused by the long carry chain to a single AND gate. This significantly reduces the overall delay in the counter design. Furthermore, the carry propagation period in C2 is much shorter than the duration of the PEN1 signal generated by C1. Overall, this novel counter architecture effectively addresses the challenges associated with large counter designs by partitioning the counter into sub-blocks, utilizing pre-scaling, and employing backward carry propagation techniques. These design choices result in improved efficiency and reduced delay, enabling higher performance for practical counter sizes.

The proposed N-bit counter introduces a simpler approach compared to the previous design [13], utilizing only three sub-counters instead of multiple ones. The partitioning of the counter is less complex, reducing the number of sub-counters involved. The delay in carry propagation within sub-counter C2 is determined by the combined delays of the AND gate, XOR gate, and D flip-flop with loading delay. The fan-out effect on the first bit is insignificant due to its small size (n-1). By setting the clock period to a minimum and incorporating a D flip-flop with a setup time, the carry propagation delay in C2 is generally faster than the duration of PEN1 generated in C1, which spans two clock cycles. The clock period is mainly influenced by the least significant sub-counter, C1. To mitigate the issue of large fan-out nodes, redundant 1-bit Johnson counters can be duplicated. The number of redundant Johnson counters, denoted as m, is determined by the equation $m = (N - n)/L$, where $L$ represents the number of input ports of a flip-flop. Typically, for $N$ ranging from 8 to 128 bits, $m$ remains below 8. While the redundancy introduces additional complexity, it only affects a small portion of the overall counter, with a maximum of 8 redundant flip-flops. To maintain the counting rate, the fan-out of a Johnson counter is limited to 16 based on extensive simulations. All Johnson counters receive the same driving signal, generating m identical PEN2 signals. Each PEN2 signal is evenly distributed and drives up to L flip-flops in the subsequent sub-counter, C3.

4 Conclusion

A novel synchronous binary counter has been developed to achieve a nearly constant delay for practical-sized counters. This design addresses the issues of large fan-out nodes and unwanted propagation delays. The proposed counter design utilizes backward carry propagation and redundant 1-bit Johnson counters. It achieves efficient functionality with a minimal number of flip-flops, slightly exceeding the counter size. Moreover, the counter is capable of operating at a stable frequency of 2GHz, exhibiting minimal dependence on the counter size. In comparison to existing methods, the proposed counter demonstrates a 25% improvement in performance.

References