A Programmable Self-Adaptive Digital Frequency Multiplier

SYED MASUD MAHmUD, MEMBER, IEEE, SUBRAMANIAM GANESAN, SENIOR MEMBER, IEEE,
ANDRZEJ RUSEK, SENIOR MEMBER, IEEE, AND MICHAEL L. HILLIS

Abstract—The existing self-adaptive frequency multipliers work well if the master clock frequency is an integral multiple of the input signal frequency, otherwise they lose an output pulse after a certain interval of time. The frequency of this missing pulse could be as high as half of the input frequency. Existing self-adaptive frequency multipliers do not have the programmable features, that means the multiplication factor can not be changed without doing some major changes in the hardware. This paper explains the reason for missing an output pulse, and presents the design and implementation of a programmable self-adaptive digital frequency multiplier which doesn’t have this missing pulse problem. The errors associated with the multiplier are also discussed.

I. INTRODUCTION

Several papers like [1]–[7] have already presented different techniques of frequency multiplication, but all of them have some limitations. In some multipliers the input frequency has to be known before the multiplier is applied [1], in others the multiplication factor can not be changed without changing some parts of the hardware [2], [4].

The design of a self-adaptive frequency multiplier is presented in [2]. A frequency multiplier is called self-adaptive if it can follow the input signal over a wide range of frequencies. Self-adaptive frequency multipliers must be designed carefully, otherwise they may lose one out of $N$ output pulses after a certain number of input pulses when $f_i$ is not exactly divisible by $f_{in}$ (where $N$ is the multiplication factor and $f_i$ and $f_{in}$ are the frequencies of the master clock and the input signal, respectively).

This paper presents the design of a programmable self-adaptive frequency multiplier. The multiplication factor $N$ could be changed any time by changing the positions of a number of switches. The maximum value of $N$ is equal to $2^n$, where $n$ is the number of switches. The existing self-adaptive frequency multiplier [2] works well when $f_i$ is an integral multiple of $f_{in}$, i.e., when $f_i/f_{in}$ is an integer. It loses an output pulse after a certain interval of time when $f_i/f_{in}$ is not an integer. This paper first explains the reason for losing the output pulses and then presents an improved design.

An exact spectrum of a periodic signal could be obtained when the data window contains an integral number of samples related to the observed waveform [4]. The FFT algorithms require a data window containing $2^N$ samples. Thus, this application requires a frequency multiplier, with multiplication factor $2^N$, to generate periodic output pulses of frequency equal to an integral multiple of the frequency of a periodic input signal. Multiplication by any number is desired when an easy way of obtaining a reasonably stable frequency reference is necessary. Frequency multiplication gives $N$ multiples of the basic frequency of the signal applied to the multiplier, which could also be useful in multiplier. This could rate generators of more complex pulse generators and in frequency synthesis.

II. PRINCIPLE OF OPERATION OF A FREQUENCY MULTIPLIER

Ideally a frequency multiplier is supposed to generate $N$ equidistant output pulses during a period of the input signal. Generally a counter is used to measure the period of the input signal in terms of the master clock cycles. The counter is started at a low-to-high transition of the input signal, and then it is incremented by 1 at every master clock pulse. At the next low-to-high transition of the input signal, the content of the counter is transferred into a buffer and the counter is restarted from zero. The content of the buffer can be expressed as

$$M = f_i/f_{in} \quad (1)$$

Since the frequency multiplier has to generate $N$ output pulses during one cycle of the input signal, it has to generate an output pulse approximately after every $M/N$ master clock cycles. Let the quotient and the remainder parts of the operation $M/N$ be $Q$ and $R$, respectively. That means

$$Q = \text{INT}(M/N) \quad (2a)$$

$$R = \text{MOD}(M/N) \quad (2b)$$

where INT and MOD mean the integer and mod functions, respectively. Almost all the early frequency multipliers used to generate output pulses after every $Q$ master clock cycles [1], [5]. Since in the design of early frequency multipliers the remainder $R$ was not taken into consideration, a very high-frequency clock had to be used in order to have a small relative frequency error. This error is defined as

$$E_r = (f_{out} - f_{in})/f_{out} \quad (3)$$
where $f_{\text{out}}$ is the desired output frequency, but $f'_{\text{out}}$ is the actual output frequency generated from the frequency multiplier. Some designs of frequency multipliers take $R$ into account [2], [4]. For the same clock frequency these multipliers have smaller relative error than those of the early multipliers. The design presented in the most recent paper [2] is simple and self-adaptive, but it has the following limitations:

i) the value of $N$ has to be a power of 2,

ii) the value of $N$ can not be changed without changing different components of the hardware,

iii) one out of $N$ output pulses will be missing after a certain interval of time if $f_c/f_{\text{in}}$ is not an integer.

The design presented in this paper is also very simple and self-adaptive. This multiplier takes $R$ into account and it does not have the above limitations.

II. DESCRIPTION OF THE PROGRAMMABLE SELF-
ADAPTIVE FREQUENCY MULTIPLIER

The design presented in this paper is an extension and improvement of the design presented in [2]. The timing and synchronization circuit, shown in Fig. 1, is the same as the one used in [2]. The $m$-bit upcounter of [2] is divided into two upcounters: upcounter-1 ($n$-bit) and upcounter-2 ($m-n$ bit). Similarly, the $m$-bit buffer (buffer #1) of [2] has been divided into two buffers: buffer-$1$ ($n$-bit) and buffer-$2$ ($m-n$ bit). The timing and synchronization circuit generates a positive pulse $Q_d$ at every low-to-high transition of the input frequency. The pulse $Q_d$ loads upcounter-1 and upcounter-2 with 2 and 0, respectively. The reason for loading upcounter-1 with 2 is that this counter loses two master clock pulses before it starts counting after every low-to-high transition of the input signal. After $Q_d$ becomes low, upcounter-1 starts incrementing by 1 at every master clock pulse. An $n$-bit comparator compares the output of upcounter-1 with the multiplying factor $N$. When the content of upcounter-1 becomes equal to $N$ the signal $V$ changes from 0 to 1. At the low-to-high transition of signal $V$, upcounter-2 is incremented by 1 and upcounter-1 is loaded with 1. Upcounter-1 then starts counting up from 1. At this time upcounter-1 is loaded with 1, because it loses one master clock pulse to get initialized. When the count value of upcounter-1 again reaches $N$ the signal $V$ again changes from low to high. At this point upcounter-2 is again incremented by 1 and upcounter-1 is again loaded with 1. This process would be continued until another low-to-high transition of the input signal comes. At this transition of the input signal, first the contents of upcounter-1 and upcounter-2 will be transferred to buffer-$1$ and buffer-$2$, respectively, and then the downcounter will be loaded from buffer-$2$, upcounter-2 will be cleared and upcounter-1 will be loaded with 2. Buffer-$3$ will also be cleared at the same time. At this point the contents of buffer-$1$ and buffer-$2$ are the remainder ($R$) and the quotient ($Q$) of the operation $M/N$, respectively. After this, upcounter-1, and upcounter-2 would repeat the entire sequence of operations, and the downcounter would be decremented by 1 at every master clock pulse.

If the content of buffer-$1$ is zero, i.e., remainder is zero, then all the $N$ output pulses would be equidistant. But if the remainder is a nonzero number ($R \neq 0$) then $R$ output pulses would be one master clock cycle wider than the other $N-R$ output pulses. The correction circuit, shown in Fig. 2, evenly distributes these $R$ wider pulses over the $N$ output pulses.

The correction circuit uses a comparator to change the signal $W$ to a high level when the output ($B$) of the $n$-bit adder is greater than or equal to $N$. After $W$ becomes high the next output pulse is widened by one master clock pulse, $N$ is subtracted from $B$ and the difference is stored in buffer-$3$. The signal $W$ will become high $R$ times during a cycle of the input signal. Table I shows the values of $B$ and $W$ during different periods of the output signal for $N = 5$ and for all possible values of $R$. The timing diagrams of the different signals are shown in Fig. 3.

This frequency multiplier would work well if $f_c$ is an exact multiple of $f_{\text{in}}$, otherwise sometimes one out of the $N$ output pulses will be missing. This fact can be explained by an example assuming that $f_c/f_{\text{in}} = 52.5$ and the value of $f_{\text{in}}$ is stable. During one cycle of the input signal, upcounter-1 will count 53 clock pulses and during the next cycle of the input signal it will count 52 clock pulses. The frequency multiplier would lose the last output pulse whenever upcounter-1 will count 52 clock pulses, because during the previous input cycle it counted 53 clock pulses, and the present input cycle will end one clock cycle before the downcounter reaches the appropriate value to generate the last output pulse. Thus, the downcounter will be reloaded from buffer-$2$ one clock cycle before the last output pulse is generated. As a result, the multiplier loses the last output pulse during an input
cycle, if during this input period upcounter-1 counts one clock pulse less than the previous input cycle. Hence, if \( f_c/f_{in} = 52.5 \) the frequency multiplier will lose the last output pulse at every alternate input cycle. This fact was verified by experimental result for \( N = 5 \), \( f_c = 86.6 \) kHz and \( f_{in} = 1.65 \) kHz \( (f_c/f_{in} \approx 52.5) \). On the oscilloscope it was seen that the first 4 output pulses were stable but the fifth one was blinking. This is shown in Fig. 4, and this figure shows that the fifth pulse is not as bright as the first four pulses. This confirms that in some input cycles the last output pulse was present and in others it was absent. In general, the average rate of missing output pulses will be one missing pulse per \( p \) input cycles, where \( p \) can be expressed as

\[
p = 1/(f_c/f_{in} - M)
\]

if \( 0 < (f_c/f_{in} - M) < 0.5 \) \( (4a) \)

\[
p = 1/((f_c/f_{in} - M) - 0.5)
\]

if \( 0.5 < (f_c/f_{in} - M) \) \( (4b) \)

where \( M \) is the integer part of the operation \( f_c/f_{in} \). The average time interval between two successive missing pulses can be expressed as

\[
T_m = T_{in} * p
\]

where, \( T_{in} \) is the period of the input signal. Hence, the frequency of the missing pulses is

\[
f_m = 1/T_m = 1/(T_{in} * p) = f_{in}/p.
\]

Substituting the value of \( p \) in \( (6) \) \( f_m \) can be expressed as

\[
f_m = f_c - M * f_{in}
\]

if \( 0 < (f_c/f_{in} - M) \leq 0.5 \) \( (7a) \)

\[
f_m = f_c - (M + 0.5) * f_{in}
\]

if \( 0.5 < (f_c/f_{in} - M) \). \( (7b) \)

The value of \( f_m \) becomes maximum when \( f_c/f_{in} - M = 0.5 \) and the maximum value of \( f_m \) is equal to \( 0.5 * f_{in} \). The problem of missing an output pulse during certain input cycles exists in the existing self-adaptive frequency multiplier. The multiplier of [2] has been built and tested to verify this fact.

The problem of missing the last output pulse could be avoided if during every input cycle, upcounter-1 counts one clock pulse less than it is supposed to count. We can achieve this by loading upcounter-1 with 1 rather than loading it with 2 at every low-to-high transition of the input cycle. Hence, we can modify the multiplier by replacing the circuit in Box-A by the circuit in Box-B (see Fig. 1). Experimental results show that this modified multiplier doesn't lose any output pulse when \( f_c \) is not an exact multiple of \( f_{in} \).

In the modified multiplier most of the output pulses will be either \( Q \) or \( Q + 1 \) clock cycles wide. On an average after every \( p \) input cycles, one output pulse will be \( Q + 2 \) clock cycles wide rather than losing the pulse where \( p \) is defined by \( (4a) \) and \( (b) \).
Another version of the multiplier can be made by replacing the correction circuit of Fig. 1 by the one shown in Fig. 2. The downcounter of this new correction circuit is loaded with the remainder \( R \) at every low-to-high transition of the input cycle, and then it is decremented by 1 at every output pulse until its content becomes equal to 0. Once it reaches 0 it is not decremented anymore. If \( R \), the content of buffer-1, is nonzero then the new correction circuit widens the output pulses 2 through \( R + 1 \) by one clock cycle, and the first and the last \( N - R - 1 \) output pulses are not widened. The \( D \) flip-flop of the correction circuit is used to reduce the instantaneous relative frequency error.

IV. PERFORMANCE CHARACTERISTICS

A. Output Pulse Period

If \( f_r \) is exactly divisible by \( f_m \) then for both the correction circuits the maximum error in output pulse period is equal to one master clock period. But if \( f_r / f_m \) is not an integer then the maximum error will be equal to two clock periods.

B. Output Pulse Position

If the correction circuit of Fig. 1 is used, then during any input cycle, except every \( p \)th input cycle, the position of each output pulse is within one master clock period of the position where it is supposed to be. During every \( p \)th input cycle the precision of the position of any output pulse is within two master clock cycles where \( p \) is defined by (4a) and (b). Using (9) of [2] the maximum output frequency during each of the first \( p - 1 \) input cycles can be expressed as

\[
f_{\text{out}} = f_r / (M/N - 1/N)
\]

and using (3) of this paper, the maximum relative error is given by

\[
E_{r_{\text{max}}} \equiv 1/M.
\]

During every \( p \)th input cycle the maximum value of output frequency can be written as

\[
f_{\text{out}} = f_r / (M/N - 2/N)
\]

and the maximum relative error is given by

\[
E_{r_{\text{max}}} \equiv 2/M.
\]

If \( f_r \) is exactly divisible by \( f_m \) then (8) and (9) give the maximum output frequency and the maximum relative error during all the input cycles, respectively.

If the correction circuit of Fig. 2 is used, then the error in position of the output pulses becomes maximum for \((R + 1)\)th pulse, and this error can be expressed as

\[
E_p = T_c \times (R - (R + 1) \times R/N).
\]

The value of \( E_p \) becomes maximum when \( R = (N - 1)/2 \) and the maximum value of the positional error can be expressed as

\[
E_{p_{\text{max}}} = T_c \times N \times (1 - 1/N)^2/4
\]

\[
\approx T_c \times N/4 \text{ (for large } N). \tag{13}
\]

C. Maximum Master Clock Frequency

If the multiplier of Fig. 1 is implemented using Schottky TTL IC's then the maximum master clock frequency could be approximately 15.5 MHz.

If the correction circuit of Fig. 2 is used, then the maximum master clock frequency could be approximately 17.4 MHz.

V. PRACTICAL REALIZATIONS

The multiplier with the correction circuit of Fig. 1 was implemented using 25 IC's. Implementation of the multiplier using the correction circuit of Fig. 2 requires 21 IC's. Four switches were used to select the multiplication factor \( N \). Thus, the value of \( N \) could be anything in the range 0 to 15. When \( N \) is set to 0 the multiplication factor becomes equal to 16. The multiplier with two different correction circuits was tested for all values of \( N \) in the range 2 to 16. All the experimental findings conformed with the expected results.

VI. CONCLUSION

The design of a programmable self-adaptive frequency multiplier is presented in this paper. If the multiplier uses the correction circuit of Fig. 1, then the precision of the interval between any two successive output pulses and the position of the output pulses are within one master clock period if \( f_r \) is an exact multiple of \( f_m \). These precisions are within two master clock cycles if \( f_r \) is not an exact multiple of \( f_m \).

If the multiplier is built using the correction circuit of Fig. 2, then 12-percent higher speed can be achieved at the cost of higher pulse position error.

The comparators which generate \( V \) signal and the output pulses must be carefully designed not to have any undesired glitches at their outputs, otherwise the multiplier will not work.

REFERENCES


