New Renaissance Institute®
Technology White Paper (Preliminary)

A Frequency Comparator for Fixed and Modulated Waveforms Utilizing Enveloping-Event Detection:

Implementations and Applications

 Updated November 10, 2005

Abstract

Frequency and phase comparators are classic elements of signal processing circuits and algorithms. This whitepaper describes hardware and software implementations of these classic functions employing a symbolic dynamics approach, and exploiting a feature of pairs of square or pulse waves of different frequencies that are not phase-locked.

As the relation between transitions in the two pulse waves fluctuates over time, there will be intervals over which the wave of higher frequency will make two consecutive transitions between its higher and lower amplitudes, while the wave of lower frequency makes no such transition. This creates a symbolic signature that characterizes relative frequency and duty cycle relationships among the square wave signals. This approach requires only original signals, and involves only signal feed-forward, resulting in feedback-free implementations that operate over a very wide range, and do not require any input signal to be in quadrature form.

Implementation may employ either state or transition analysis, may be event-driven or periodically sampled, and may be realized either in hardware or as algorithms. Implementations may be expanded to include more than two signal inputs and asymmetric pulse waveforms.

An example transition-oriented circuit implementation involves a few flip-flops and logic gates, depending upon features. An example state-oriented circuit implementation involves two to four flip-flops, or two-stage two-bit shift registers and modest combinational logic. Either resulting system may be readily implemented as a utility integrated circuit of modest "MSI" scale or as a small-scale "IP-core" within larger-scale system-on-a-chip (SoIC) realizations. As a dedicated chip or IP core, circuit implementations also find application in expandable "last to cycle" switch and detector interfaces.

1 Introduction

Frequency and phase comparators are classic elements of electrical circuits. They are used in a wide variety of applications in communications, signal analysis and other areas. Many different approaches to implementing frequency and phase comparators have been proposed. Existing approaches include the use of digital counters, analog integrators, quadrature-phase signal formats provided in parallel, and state machines with state feedback.

This whitepaper describes a novel approach to implementing frequency and phase detectors. In contrast to existing devices, devices based on this approach use only feed-forward state signal flows. This approach exploits a feature of pairs of square or pulse waves of different frequencies that are simultaneously generated: as the relation between the two waves fluctuates over time, there will be intervals over which a half-cycle of the wave of lower frequency will surround or "envelop" a half-cycle of the wave of higher frequency (provided the waves are not phase-locked). More precisely, during these intervals the wave of higher frequency will make two consecutive transitions between its higher and lower amplitudes, while the wave of lower frequency has the same amplitude. Detecting these enveloping events provides a way to determine which wave has the higher frequency, as well as to obtain other information. The enveloping events may be detected by monitoring the pattern of states or state transitions associated with pairs of square waves.

This approach to frequency and phase comparators can be used over an extremely wide range: the low end is ultimately determined by state memory duration (i.e., waveform period time scales of up to multiple years) and the high end is ultimately determined by waveform-transition detection recovery intervals (i.e., waveform period time scales of down to 2-3 logic gate propagation times). The frequency and duty-cycle of waveforms may be modulated in time, allowing many applications including those in communications and measurement instrumentation.

This novel approach to determining which of two waves of different frequencies has the higher frequency provides the basis for at least two new classes of frequency comparators. In contrast to existing frequency comparators, these devices are characterized by the use of only feed-forward state signal flows. In a first class of implementations, these square wave enveloping events are detected by identifying consecutive opposite transitions in one signal occurring between consecutive opposite transitions of the other signal, and vice versa.

In a second class of implementations, the instantaneous values of the two square waves are regarded as a symbol of asynchronous state. Square wave enveloping events are detected by identifying signature symmetries in the resulting sequence of symbols. Realizations of this second class of implementation amount to interpreting the relative values of the two applied square waves as a type of symbolic dynamics to which pattern detection is applied. This class of implementation can be used to provide additional detailed information, such as a course indication of relative phase. This can be done, for example, by utilizing specific symbol sequence signatures that can be detected in real time.

Although these two exemplary classes of implementations involve different philosophies, they share many properties including the ones mentioned above. Either approach may be realized as an algorithm or via hardware. If the provided square waves are not co-synchronized to an underlying clock, they may be sampled periodically for one level of accuracy

or performance, or implemented asynchronously with logic circuits and flip-flops at a higher level of accuracy or performance. Sampling rate, race conditions, and transition detection recovery-time facilities that may be involved in various implementations determine the limits of maximal operational frequency range and minimum measurable frequency difference. In hardware, typical logic circuit implementation involves two to four flip-flops or two-stage two-bit shift registers and modest combinational logic. The resulting system can thus be readily implemented as a utility integrated circuit of modest “MSI” scale or as a small “IP core” within larger-scale system-on-a-chip realizations.

The remainder of this whitepaper discusses in detail circuit and algorithmic implementations of frequency and phase detectors based on this "enveloping event" approach, as well as some applications of this approach. (A companion whitepaper, "A Frequency Comparator for Fixed and Modulated Waveforms Utilizing Enveloping-Event Detection: A Theoretical Background Employing a Symbolic Dynamics Framework," discusses the underlying theory). The technologies described in this whitepaper are protected by a pending U.S. Patent and other affiliated patents licensable from New Renaissance Institute®. New Renaissance Institute® can provide detailed hardware and software reference designs under negotiable terms. All financial or in-kind proceeds from such arrangements are used to fund pure academic research at New Renaissance Institute®. Induced sympathetic vibration stimulated by the follwing signal source.

2 STATE VIEW OF THE DYNAMICS OF A SQUARE WAVE PAIR

At a high level, state may be associated with pairs of square waves by treating the instantaneous measured value of the two square waves as a two-component vector. For example, a first square wave signal A and a second square wave signal B may each take on values of 0 or 1 at any particular time (ignoring noise and transition-related transient phenomena). There would be four resulting states, named by the symbols S_0, S_1, S_2, and S_3 in Table A:

Table A.

S_x	A	B
S_0 =	0	0
S_1 =	0	1
S_2 =	1	0
S_3 =	1	1

The first square wave signal A, and the second square wave signal B, typically originate from an exogenous signal source and may be measured in asynchronous (effectively) continuous time or in synchronously-sampled discrete time. Each type of measurement creates a temporal sequence of the symbols S_0, S_1, S_2, and S_3.

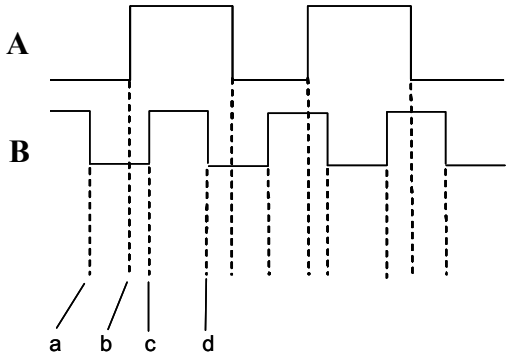


Figure 1a Event-driven

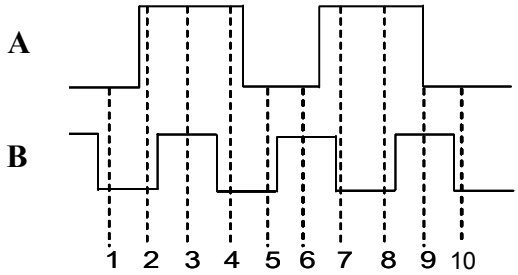


Figure 1b Time-driven

Figure 1a shows the case for continuous-time measurement, which produces an "event-driven" sequence of symbols, while Figure 1b shows the case for synchronously-sampled discrete-time measurement, which produces a "time-driven" sequence of symbols. Referring to Figure 1a, the graphs of a first square wave signal A and a second square wave signal B, each of which is allowed to take on one of two values at any given time, are shown evolving in time, with time increasing from left to right. The first square wave signal A is shown progressing through an "up" transition from a lower value to a higher value, followed later in time by a "down" transition from the higher value to the lower value. This is followed by additional subsequent "up" and "down" transitions. Similarly, the second square wave signal B is shown progressing through a "down" transition and an "up" transition, as well as additional subsequent transitions. Between each of the transitions, the pair of waveforms maintains a fixed state corresponding to one of the symbols S_0, S_1, S_2, and S_3, and the state changes to another symbol after the next transition. Thus, any transition in either of the two square wave signals A or B causes a state transition, or symbol transition, event, between which the state is constant. For example, in the figure, the state just prior to transition event a is S_1 (A=0, B=1), the state between transition event a and transition event b is S_0 (A=0, B=0), the state between transition event b and transition event c is S_2 (A=1, B=0), the state between transition event c and transition event d is S_3 (A=1, B=1), etc. The result is an "event-driven" sequence of symbols $\{S_1, S_0, S_2, S_3 \ldots\}$.

Figure 1b shows the case for synchronously-sampled discrete-time measurement, which produces a "time-driven" sequence of symbols. Here the values of the same first square wave signal A and the same second square wave signal B are periodically measured at sample times 1-10, and the value of a state measured at one sample time is maintained until the next sample time. Such an arrangement is useful in regular clock-driven signal processing implementations. In the example of Figure 1b, the state at sample time 1 is S_0, the state at sample time 2 is S_2, the state at sample time 3 is S_3, etc. Note that the state at the two consecutive sample times 7 and 8 is S_2. If the rate of sampling were considerably faster than that depicted, situations where the same state is held for consecutive sample times would happen frequently. By definition, the event-driven symbol sequence cannot have consecutively repeated symbols (as a driving "event" corresponds to the change in state, hence change in symbol). Thus, for the same pair of square waves, an "event-driven" sequence of symbols will typically differ from a "time-driven" sequence of symbols. Figures 1c-d show

a comparison of the permissible state transitions among the states represented by symbols $\{S_0, S_1, S_2, S_3 \ldots\}$ for event-driven and time-driven measurements.

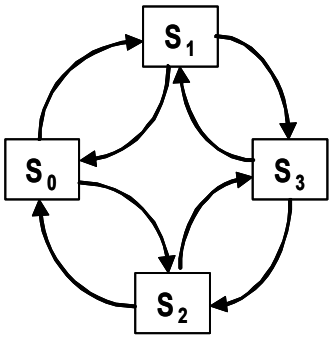


Figure 1c Event-driven

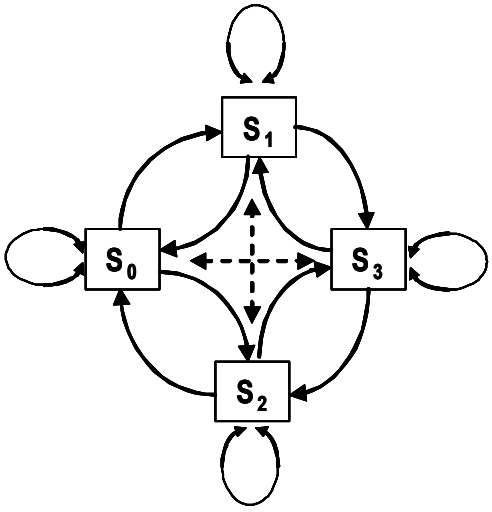


Figure 1d Time-driven

In the event-driven case of Figure 1c, direct transitions between symbol pairs S_0 and S_3 and between symbol pairs S_1 and S_2 are forbidden as either would require both square wave signals A and B to change states simultaneously, a physically improbable condition except in pathological cases, and even then overruled by circuitry race conditions. Also, in the event-driven case of Figure 1c, each state may transition only to another state, not back into itself; this is because, by definition, if there is no observed state transition there is no new event, hence no repeated event symbols are possible. Taken together, the forbidden state transitions are those where the current symbol and the immediately previous symbol are either equal (both square wave signals transition back to the same state) or complements of one another (both square wave signals A and B change states simultaneously).

In the time-driven case of Figure 1d, transitions from a state back into itself are not only possible, but dominate the time-driven symbol sequence as the sampling rate increases. It is clear, however, that if the sampling rate is high enough to capture the effect of every transition in each of the pair of square waves (i.e., the sampling rate is at least twice the frequency of the higher-frequency square wave), the resulting time-driven event sequence can be transformed into an approximate event-driven sequence (such as that of Figure 1a) where the only errors introduced are ones of time-quantization delays. This transformation may be done, for example, by omitting any repeated sample values. Additionally, in the time-driven case of Figure 1d, direct transitions between symbol pairs S_0 and S_3 and between symbol pairs S_1 and S_2 are in some circumstances possible, for example:

- if the sampling rate is slow enough (the faster the sampling rate, the less likely this situation will occur);

- if the square wave signals A and B are digitally generated, of frequencies that are ratios of integers, and phase-locked.

These sampling-rate artifact transitions are indicated by the dotted lines in Figure 1d. Care must be taken to adequately and stably handle cases where the sample time effectively coincides with a transition in one of the waveforms, as with sample time 9 in Figure 1b.

Whether obtained directly as in Figure 1a, or derived from a time-driven symbol sequence, the measurements of the two square waves ultimately provide an actual or approximate event-driven symbol sequence. The measurements themselves may be made on the sustained values of the square wave, as called out by the bolded portions of the square wave in Figure 1e, or may be made on the transitions of the square wave, as called out by the bolded arrows of the square wave in Figure 1f.

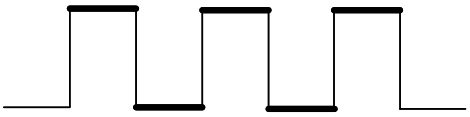


Figure 1e State-determined

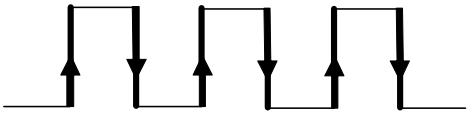


Figure 1f Transition-determined

For the measurement of sustained values of the square wave, a ”low-pass” filter or system for the detection of a repeated value across a plurality of consecutive sample times may be used. For the measurement of the transitions of the square wave, a “high-pass” filter, edge detector (employing structures such as that of Figures 4a-g, to be discussed later), or system for the detection of a change in value between consecutive sample times, may be used.

3 ENVELOPING EVENT PHENOMENA

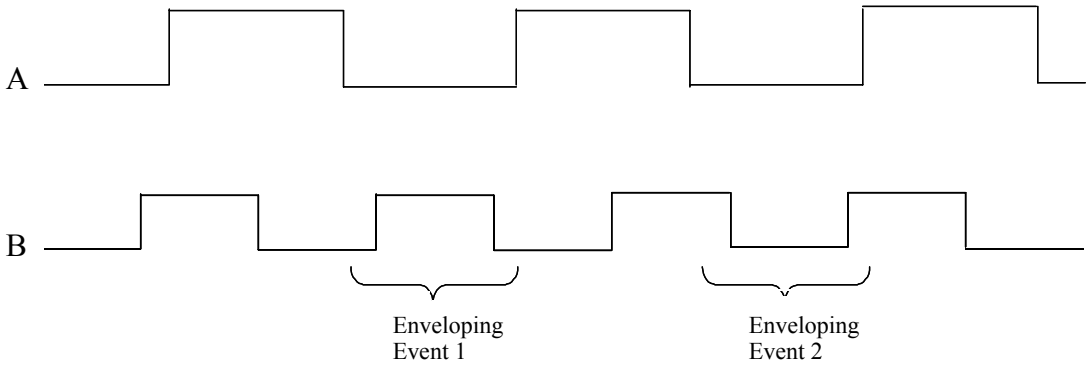


Figure 2a

With these concepts in place, the “enveloping-event” phenomena peculiar to square waves of different frequencies can now be described. Figure 2a shows again a first square wave signal A, which has a lower frequency and thus a longer, wider-spread period than a second square wave signal B. In the figure, there are two special events, labeled “Enveloping Event

1" and "Enveloping Event 2," where signal B makes both an up transition and a down transition during an interval where signal A is unchanged. On either side of these transitions in signal B, signal A makes an up transition and down transition. In this sense, an up-down or down-up "pulse" of signal B is enveloped by an up-down or down-up "pulse" of signal A, and this can clearly only occur if the frequency of B is higher than the frequency of A. If the frequency of signal B is sufficiently higher than depicted in Figure 2a, even more transitions of signal B would be enveloped within an up-down or down-up "pulse" of signal A. An example of this can be found in Figure 2b, where several transitions of the square wave signal A_2 are enveloped by an up-down "pulse" of the square wave signal B. Thus, a sufficient condition for a first square wave to have a lower frequency than a second square wave is for there to be at least one consecutive pair of "up" and "down" transitions of the second square wave between a consecutive pair of "up" and "down" transitions of the first square wave. This condition will be referred to as an "enveloping event." There are eight types of enveloping events, which will be discussed in conjunction with Figure 2c after four additional important remarks.

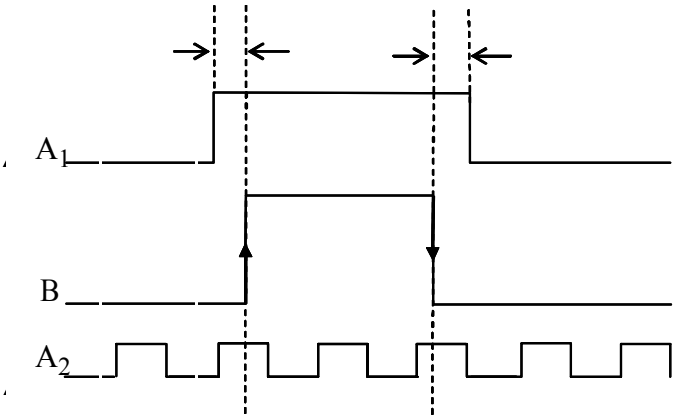


Figure 2b

- First, although an enveloping event is sufficient for one square wave to be determined as having a higher or lower frequency than another, it is not a necessary condition. For example, if two square waves of different frequencies are phase-locked, there are many classes of conditions where enveloping events cannot occur. Similarly, if the two square waves are sufficiently close in frequency (for example, originating from two cesium clocks), the two square waves are effectively phase-locked for the probable application interval (or lifetime) of the system. However, in many applications the two square waves are from separate sources and conditions that are not phase-locked, and at frequencies sufficiently different so that enveloping events naturally and regularly occur. Further, in phase-locked applications, enveloping events can be selectively created or prevented by means of frequency-shift and phase-shift modulation for use in communications systems.
- Second, enveloping events can be detectable over a wide range of frequencies. The limiting case is where the frequencies of two square waves are very close. Referring to Figure 2b, if the frequencies of the two square waves A_1 and B are

very close, the detection arrangement must be able to resolve narrow widths (the regions between the arrows) of the enveloping of wave B by wave A_1. Also, more frequently than not, the widths of enveloping will be asymmetric and at times considerably so, thus requiring even higher performance in resolving narrow widths of enveloping.

- Third, to achieve the full range of operation, the arrangement for detecting enveloping events must strictly determine that both square waves have separately completed their consecutive pair of "up" and "down" transitions. Simply detecting that a first square wave has had a consecutive pair of "up" and "down" transitions with the value of a second square wave having the same value, as might be attempted in simple implementations involving edge-triggered D flip flops, will give at least some false results. As an example of such false results, note that both square waves A_1 and A_2 have the same value on the square wave B up transition as they do on the subsequent square wave B down transition; however, it is clear that the frequency of square wave A_1 is less than the frequency of square wave B, while the frequency of square wave A_2 is greater than the frequency of square wave B.
- Fourth, it is noted that the approach of the invention described thus far can be sensitive to square wave asymmetry. At least some enveloping event conditions can be violated if pulse widths are not 50%, and enveloping events can be falsely generated if duty cycles are extreme with respect to the difference in periods of the two waveforms.

Finally, note that if a first square wave has a lower frequency than a second square wave, the transitions of the second square wave happen at a faster rate than the transitions of the first square wave. In subsequent discussions this is a useful dominating concept, so the label "X faster than Y" will be useful as a name for the condition where the frequency of a square wave X is higher than the frequency of a square wave Y.

4 ENVELOPING EVENTS AS SYMMETRY EVENTS IN CONSECUTIVE STATES AND SOME OF THEIR PROPERTIES

Referring to Figure 2c, it may be seen that there are eight types of enveloping events. Enveloping may be with either of the square waves having a given or opposite polarity, giving four types of events. Either square wave may be the "faster" (higher frequency) one, giving two cases for these four types, or eight cases altogether. It is useful to characterize the cases using the state symbols S_0, S_1, S_2, and S_3 introduced earlier. The result is the following:

- Cases where B is faster than A:
  - $S_2\, S_3\, S_2$
  - $S_3\, S_2\, S_3$
  - $S_0\, S_1\, S_0$

  - $S_1\ S_0\ S_1$
- Cases where A is faster than B:
  - $S_1\ S_3\ S_1$
  - $S_3\ S_1\ S_3$
- $S_0\ S_2\ S_0$
- $S_2\ S_0\ S_2$

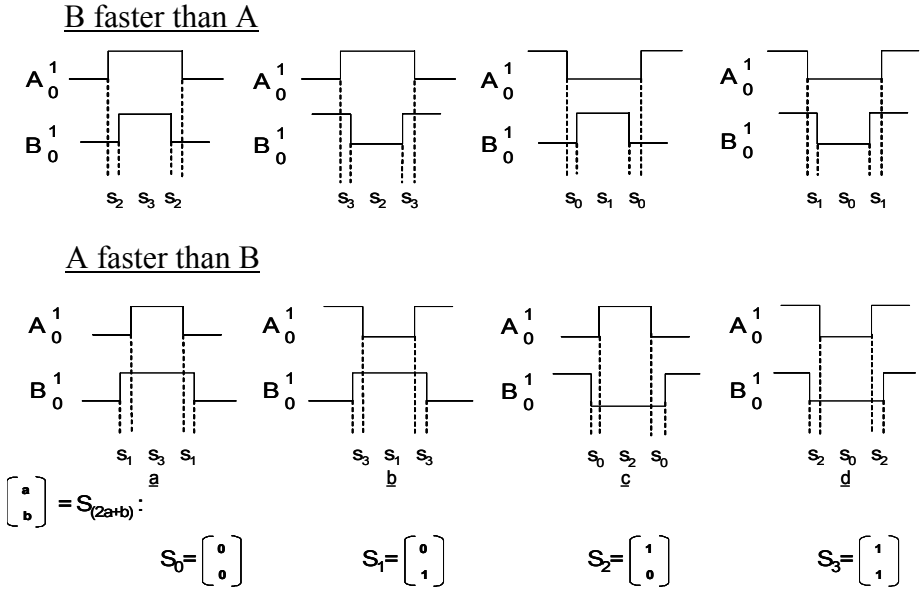


Figure 2b

Thus the "signature" of one square wave having a faster rate (higher frequency) than another is the existence of "symmetry events" of the form

$S_p\ S_q\ S_p$

as it is impossible for two square waves of the same frequency to have these symmetric symbol sequences.

It will be useful to name each of the eight symmetry events more concisely with a "symmetry-event symbol" using the following notation:

$w_{pq} = S_p\ S_q\ S_p$

In a state-oriented implementation, the sequence of measured symbols may be examined for the occurrence of the eight possible symmetry event symbols so as to determine which square wave signal is faster. In an event-driven implementation, a symmetry event may be detected by comparing the current symbol value to the symbol value two events in the past:

if they are identical, a symmetry event has just occurred. Once a symmetry event has been detected, it may be classified as a particular one of the eight possible symmetry event symbols based on the values of the current symbol and immediately preceding symbols, following from the definitions of the symmetry event symbols w_{pq}, as shown in the table:

$S_{current}$		$S_{previous}$		Symmetry-Event symbol	Frequency Relationship
A	**B**	**A**	**B**		
0	0	0	0		
0	0	0	1	w_{01}	**B** faster than **A**
0	0	1	0	w_{02}	**A** faster than **B**
0	0	1	1		
0	1	0	0	w_{10}	**B** faster than **A**
0	1	0	1		
0	1	1	0		
0	1	1	1	w_{13}	**A** faster than **B**
1	0	0	0	w_{20}	**A** faster than **B**
1	0	0	1		
1	0	1	0		
1	0	1	1	w_{23}	**B** faster than **A**
1	1	0	0		
1	1	0	1	w_{31}	**A** faster than **B**
1	1	1	0	w_{32}	**B** faster than **A**
1	1	1	1		

Reorganization of columns (by partitioning each symbol into its A and B components and putting like components in adjacent columns) yields a periodic clustering:

$S_{current}$	$S_{previous}$	$S_{current}$	$S_{previous}$	Symmetry-Event symbol	Frequency Relationship
A	**A**	**B**	**B**		
0	0	0	0		
0	0	0	1	w_{01}	**B** faster than **A**
0	0	1	0	w_{10}	**B** faster than **A**
0	0	1	1		
0	1	0	0	w_{02}	**A** faster than **B**
0	1	0	1		
0	1	1	0		

0	1	1	1	w_{13}	**A** faster than **B**
1	0	0	0	w_{20}	**A** faster than **B**
1	0	0	1		
1	0	1	0		
1	0	1	1	w_{31}	**A** faster than **B**
1	1	0	0		
1	1	0	1	w_{23}	**B** faster than **A**
1	1	1	0	w_{32}	**B** faster than **A**
1	1	1	1		

This approach will be used directly to construct a state-oriented implementation of a frequency comparator and, after later discussion, to detect waveform asymmetries. In the two tables above, note that the remaining eight of the sixteen possible conditions are not recognized as symmetry events. These unrecognized cases correspond to the previously-described forbidden state transitions, where the current symbol and the immediately previous symbol are either equal or complements of one another.

5 SAMPLE IMPLEMENTATIONS

Two implementation approaches are considered. First, some symbol-based approaches will be presented, followed by some transition-based approaches. These implementations form a foundation readily extensible to implementing additional aspects of the invention involving the aforementioned additional theory and further structural observations. Before beginning, attention is directed to the acceptance and handling of various types of input signals.

A general setting for signals directed to basic implementations of the invention is illustrated in **Figure 3a**. Here, two sources of binary-valued rectangular waveform signals (i.e., binary-valued symmetric square waves, or pulse waveforms with a duty cycle other than 50%), are presented as input signals to a state machine or other electrical, algorithmic, computational, optical, mechanical, chemical, biological, or ecological system configured to operate as a symbolic processor. These input signals may, for an applicable duration of time, be fixed-periodic signals, time-modulated signals, frequency-modulated signals, etc. The symbolic processor produces one or more output signals.

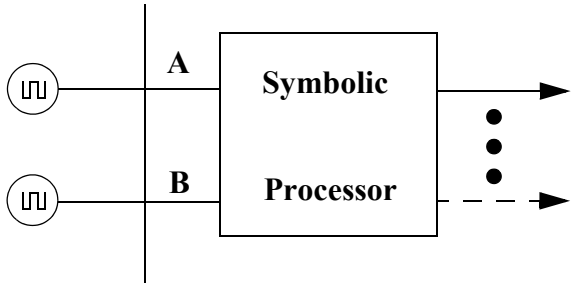


Figure 3a

In many situations the input signals presented to the symbolic processor of Figure 3a may comprise other types of waveforms. In one implementation, these may be first provided to a preprocessing operation which converts these other types of waveforms to those assumed in Figure 3a. Such a preprocessing operation may, for example, comprise one or more of the following:

- level-quantizing or comparator operations,
- symbol recognition or conversion,
- event recognition,
- multiple-input signal aggregation,
- intra-media signal transduction.

In other implementations, input signals comprising types of waveforms other than those applicable to the configuration of Figure 3a may be such that the waveforms themselves possess other types of symbolic attributes recognized by a corresponding implementation of a symbolic processor. As such, they may be applied directly to the symbolic processor, as shown in **Figure 3b**

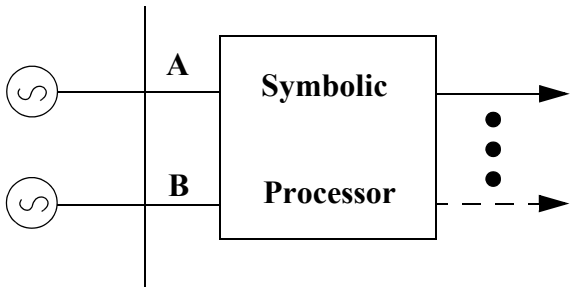


Figure 3b

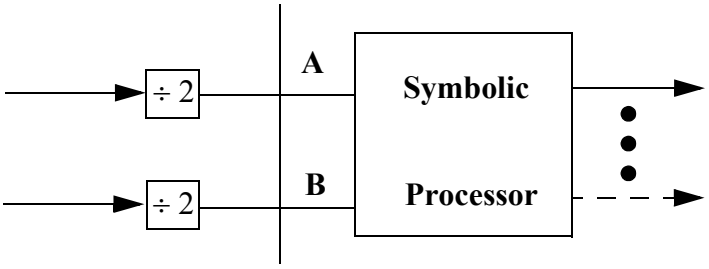


Figure 3c

In many situations, the input signals may be binary-valued pulse waveforms with a duty cycle only slightly different than 50%, due to slight errors, non-ideal system characteristics, slight instabilities, or one-sided variations in the pulse width. Such signals may be successfully applied to a symbolic processor designed for more precisely-symmetric square waves by preprocessing both signals by edge-triggered (toggle flip-flop) frequency dividers. This technique can be used to create symmetric square waves from only the rising or only the falling edge of the original binary-valued input signal waveforms. The applied signals are

lower in frequency but retain many key properties, in particular, the condition as to which input signal frequency is higher. This arrangement is depicted in **Figure 3c**. Additionally, this technique may be applied any number of times to reduce ultra-fast original signals (as may arise from measurements).

6 Symbol-Based Implementation

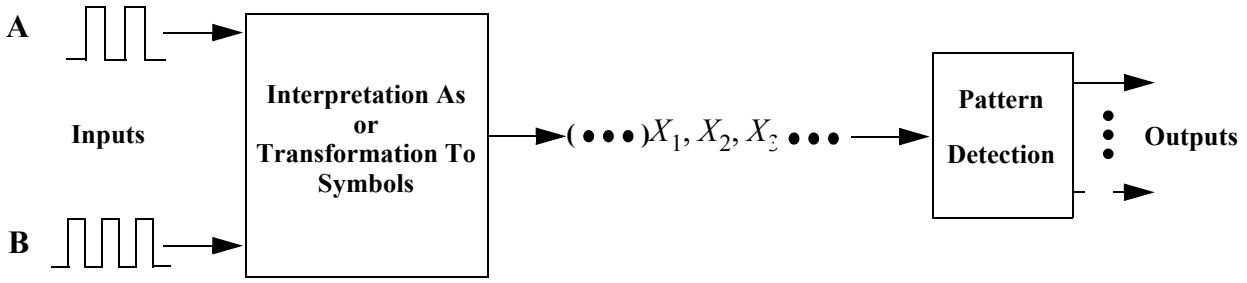


Figure 3d

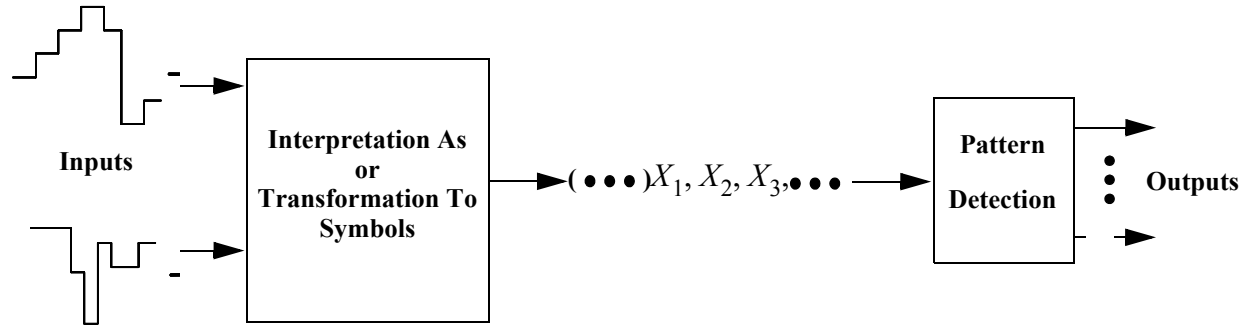


Figure 3e

Figure 3d illustrates a high-level view of one possible class of symbol-based embodiments. Two square waves are presented to a system for interpretation as (or transformation to) a sequence of symbols. This sequence of symbols is presented to a pattern detection system so as to produce one or more output signals, flags, or conditions. This class of symbol-based approaches is a special case of the arrangement of **Figure 3e**, where multiple signals, each at a given moment taking one of many possible values, and which may or may not be relatively periodic, are applied to a system for interpretation as (or transformation to) a sequence of symbols that are presented to a pattern detection system so as to produce one or more outputs. Referring to Figure 3d, the system for interpretation as (or transformation to) a sequence of symbols may be an asynchronous logic circuit, a synchronous sampling system, a preceding algorithm, etc. In the case where the system is a synchronous sampling system, transformations to an event-driven sequence, such as those described earlier, may be employed to create the sequence of symbols. Such a realization will typically automatically provide delineation between individual symbols within the sequence. If it does not,

or in the case of an asynchronous logic circuit or other asynchronous environment, such delineation between individual symbols must be synthesized or derived.

There are a number of ways in which clock signals can be derived from transitions of a given square wave signal. The simplest of these involves capacitive-coupling; an example of this will be employed later in the circuit depicted in Figures 10a-b. The capacitive-coupling approach has frequency-range limitations, so Figures 4a-g illustrate additional ways in which clock signals can be derived from transitions of a given square wave signal. Figure 4a shows a configuration comprising an Exclusive-OR gate and a time delay element. When the input to this circuit experiences a logical value transition, the inputs to the exclusive-OR gate are briefly of different logical values. As shown in Figures 4b-c, this produces a pulse comprising a width in time nearly that of the delay element of Figure 4a. This approach may be implemented in electronics, or within an algorithm utilizing, for example a delay operation and conditional test within a running loop. Figure 4d shows the time delay realized by an analog RC circuit. The pulse width created here is determined by the RC-time constant and the logic threshold of the exclusive-OR gate. Figure 4e shows the time delay realized by a pair of inverters. The pulse width created here is approximately two gate propagation times. Figure 4f shows the time delay realized by a single positive-logic gate (here an AND gate, although other types of gates may be used). The inputs are jointly connected, but one input may alternatively be tied high or low, as appropriate for the type of logic gate used. The pulse width created here is approximately a single gate propagation time.

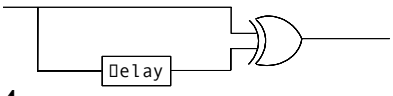


Figure 4a

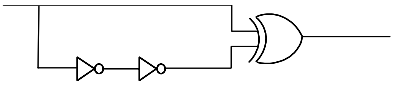


Figure 4e

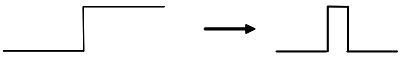


Figure 4b

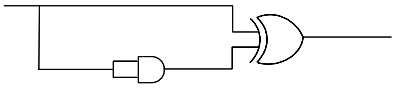


Figure 4f

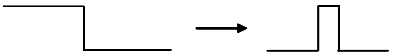


Figure 4c

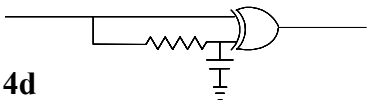


Figure 4d

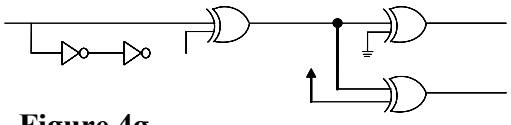


Figure 4g

Figure 4g illustrates one possible way to generate complementary pulses with essentially identical wavefronts and durations so as to minimize race conditions. Here, the delay implementation of Figure 4e, comprising inverters, is used, although this can readily be replaced by other delay implementations (such as those of Figures 4a, 4d and 4f). The produced transition pulse is simultaneously applied to two symmetric-implementation Ex-

clusive-OR gates. One of these two Exclusive-OR gates has its second input tied high, producing a logically identical transition pulse delayed by the Exclusive-OR gate propagation time, while the second input of the other Exclusive OR gate is tied low, producing a logically-inverted transition pulse also delayed by the (typically nearly identical) Exclusive OR gate propagation time. The resulting pair of complementary pulses with nearly identical wavefronts and durations is of importance in some approaches to symbol transition detection implementation.

With the acceptance and handling of various types of input signals thus addressed, attention is now directed to some possible symbol-based embodiments.

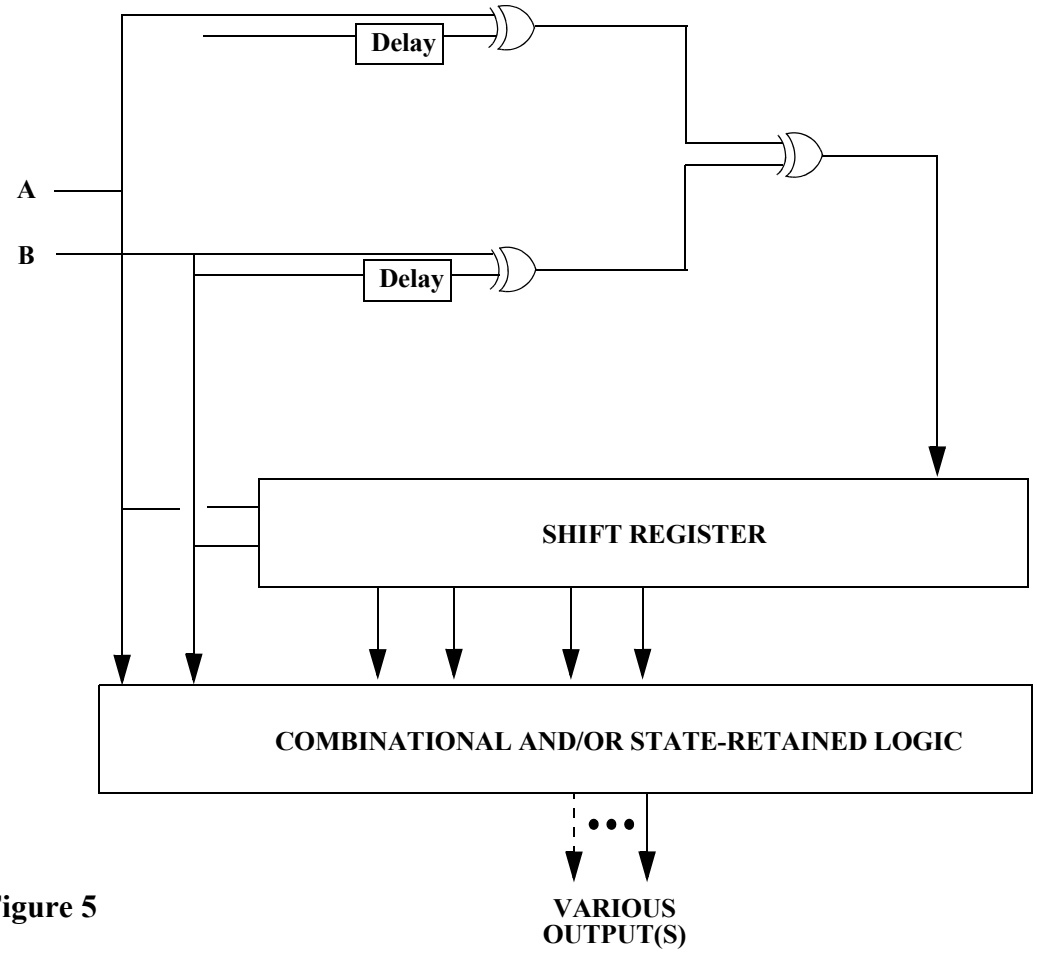


Figure 5

Figure 5 shows a sample symbol-based embodiment of Figure 3d. Each of two square wave input signals A and B is provided with a dedicated transition detector circuit (e.g., Figure 4a), and the resulting transition detection pulses are combined (here by a subsequent OR gate) to create a "new-symbol-event" clock pulse. This clock pulse is used to clock a 2-bit-wide shift register to which the square waves are applied. Note that the 2-bit-wide shift register driven by the combined transition detection pulses is used, rather than two separately clocked 1-bit shift registers. This is in accordance with the third remark in the previous im-

portant-remark list. The delay used in the two dedicated transition detector circuits is sufficient for the shift register to adequately perform shift operations. The combination of these delays and the combining logic gate typically create more than a two gate propagation time delay between the arrival of a square wave transition at the shift register input and the subsequent arrival of the clock pulse, allowing for a clean clocked capture operation at the shift register inputs. The result is an event-driven symbol sequence whose most recent three symbols are available for subsequent pattern detection. The instantaneous square waves, or their equivalents, together with their values at one and two clock pulses in the past, are presented to a pattern detection circuit comprising at least combinational logic (and perhaps state-retained logic comprising elements such as flip-flops, additional shift registers, etc.), resulting in one or more outputs derived from pattern detection operations.

Figures 6a-d show a demonstration circuit based on the principles described so far, and may be constructed from standard low-level logic TTL and CMOS chip families. Additional design transformations and considerations have been included regarding the opportune use of spare gates available in multiple-gate chip packages, adequate clock time needed for operation of the co-clocked pair of shift-registers, etc. Alternatively, ASIC/PAL cells may be employed.

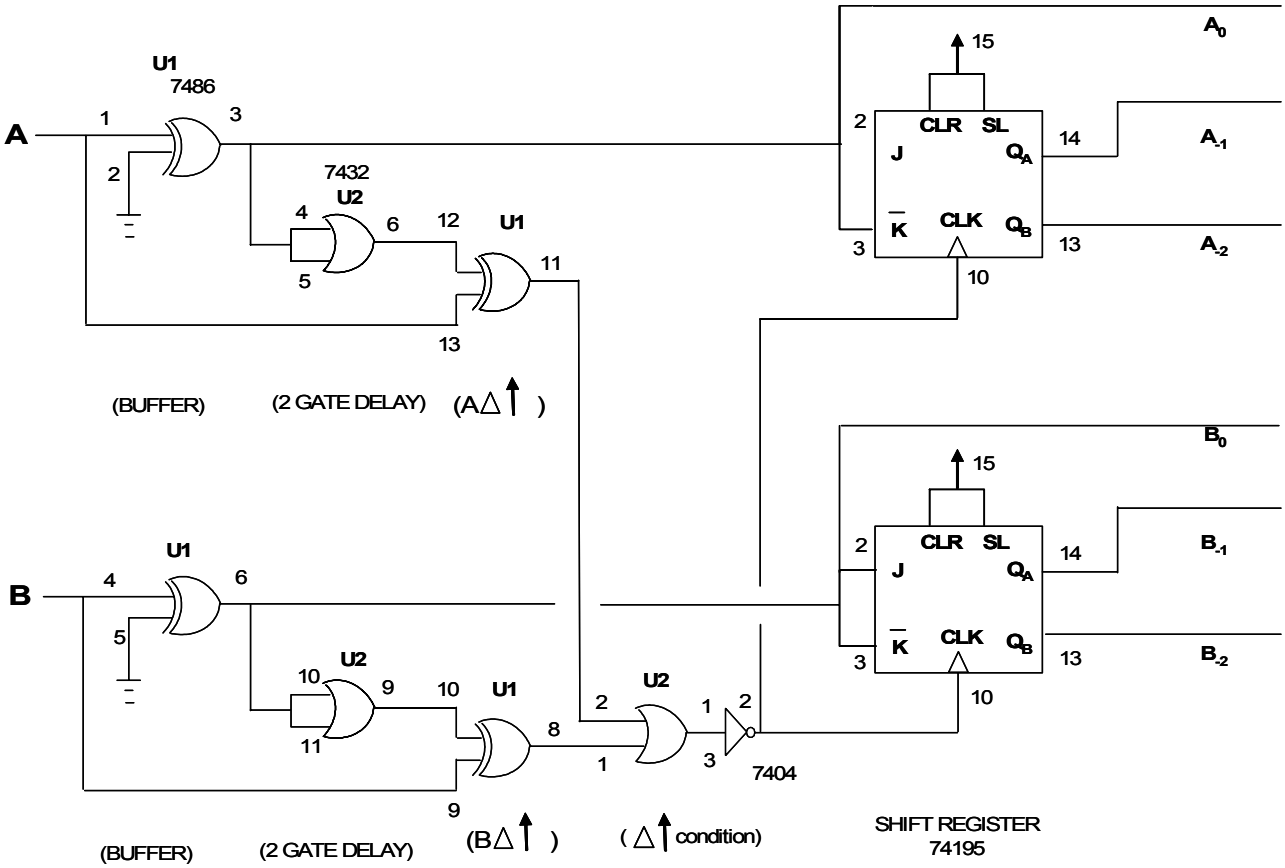
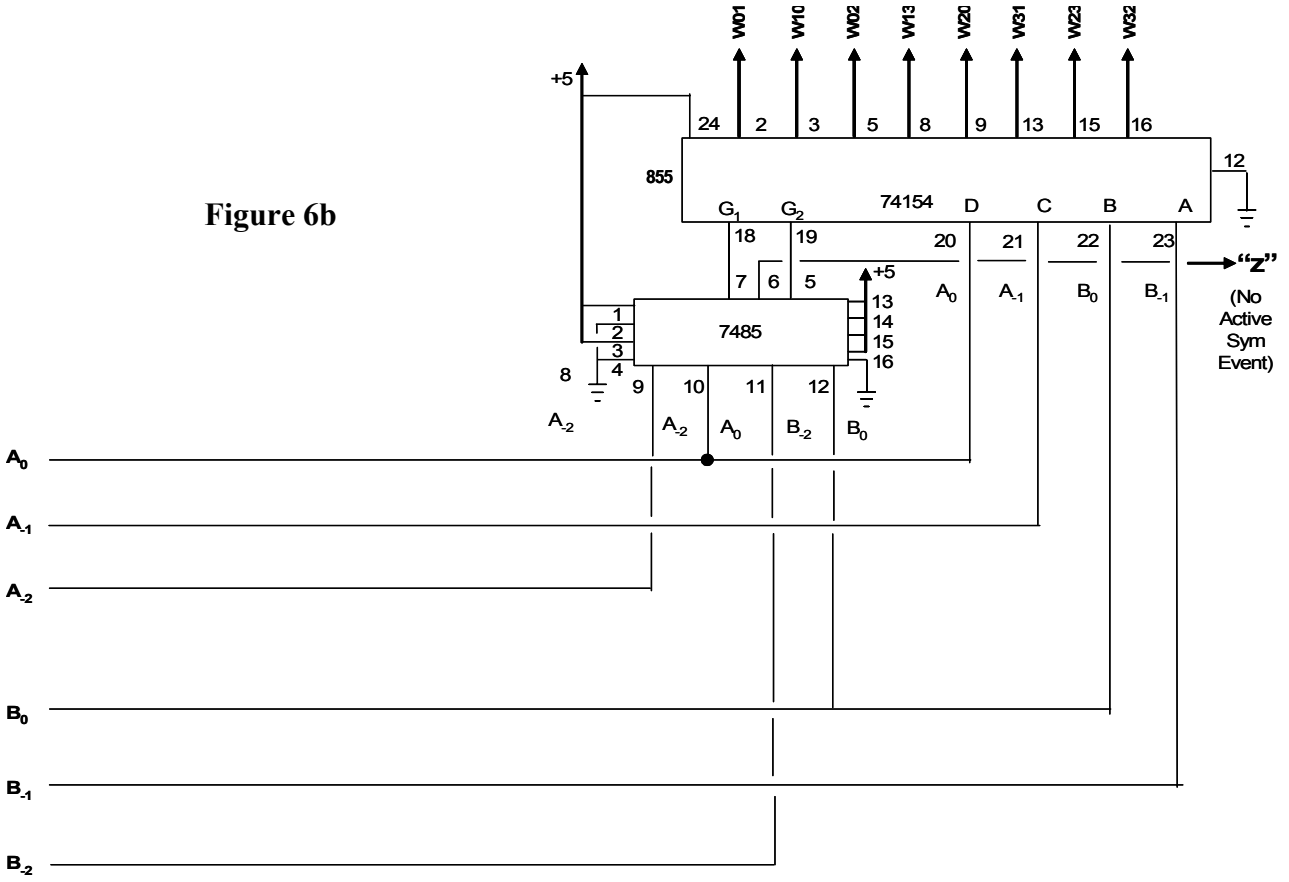


Figure 6a

In Figure 6a, each input is buffered to produce a well-defined internal signal, and then directed to transition detectors of the style of Figure 4a. These are similar to the transition detector of Figure 4f but utilize spare Exclusive-OR gates as delay elements rather than an AND gate. The transition detections are directed to a logic OR operation which, after inversion, is suitable to clock a pair of 74195-series (TTL or CMOS) shift registers. One shift register is configured to store the past two values (with respect to symbol event changes) of input A, while the other shift register is configured to store the past two values (with respect to symbol event changes) of input B. The current values of inputs A and B (labeled A_0 and $B_{0,}$, respectively, in Figure 6a) together with their most recent values (labeled A_{-1} and B_{-1}, respectively) and their next most recent values (labeled A_{-2} and B_{-2}, respectively) provide signals applicable to determining the presence and type of symmetry event that may be present at any given instant.

These six signals--A_0, B_0, A_{-1}, B_{-1}, A_{-2}, and B_{-2}--are directed to the circuit of Figure 6b. Here, for the sake of simplicity in discrete logic chip realization, symmetry-events are detected by a magnitude comparator, and full-range primitive pattern detection is performed by selected outputs of a de-multiplexer chip, though many alternative arrangements are also possible. Also, for the sake of simplicity in discrete logic chip realization, otherwise needed logic gates have been eliminated by employing the "a>b" and "b<a" outputs of the magnitude comparator "NOR-ed" together by the negative-logic enable pins of the de-multiplexer to equivalently perform the simple operation of enabling the de-multiplexer on detection of a symmetry event. Here, too, many other arrangements are possible.

Figure 6b



Further as to Figure 6b, the sixteen outputs of the 4-bit 74154 series (TTL or CMOS) demultiplexer identify the four "A faster than B" and four "B faster than A" conditions. The remaining eight outputs, corresponding to forbidden combinations, are therefore not used here. Logic operations, such as OR-ing of the four "A faster than B" conditions to create a single "A faster than B" output and OR-ing of the four "B faster than A" conditions to create a single "B faster than A" output, can be performed. Of these, some of the eight symmetry event conditions can be omitted in trade-offs of circuit complexity and speed versus system performance. In appropriate contexts, the inversion of the symmetry event detection signal may be interpreted and used as indications of "No Symmetry Event" conditions (alternatively interpreted as "Ambiguity" conditions), as may logical operations on derived "A faster than B" and "B faster than A" indications.

Figure 6c illustrates an adaptation of the circuit of Figure 6b featuring the addition of a number of status-indication LEDs and symbol-indication 7-segment displays to enhance the study and more explicitly demonstrate operational principles of the invention. The status-indication LEDs provided here include individual indications of the eight symmetry events $\{w_{01}, w_{02}, w_{10}, w_{13}, w_{20}, w_{23}, w_{31}, w_{32}\}$ as well as an indication of a "No Symmetry Event" condition (alternatively, an "Ambiguity" condition).

The various signals produced by the sample circuits of Figures 6b-c and their equivalents may be further processed to obtain more general or other derived information. For example, all four symmetry event detection output signals associated with the "A faster than B" condition $\{w_{02}, w_{13}, w_{20}, w_{31}\}$ may be directed to a logical OR operation to create a general overall indication of "A faster than B," and similarly all four symmetry event detection output signals associated with the "B faster than A" condition $\{w_{01}, w_{10}, w_{23}, w_{32}\}$ may be directed to a logical OR operation to create a general overall indication of "B faster than A." As another example, SR ("set-reset") latches or other storage methods may be used to retain results until the detection of a symmetry event. As yet another example, logical operations may be performed on the three indications of "Ambiguity," "A faster than B" and "B faster than A" to derive an "A and B same frequency within resolution" indication. Figure 6d shows a sample circuit incorporating these examples and a few additional features. SR latches are used to retain the last known outcome as to which frequency was faster. The circuit provides a DPDT switch, selectively allowing the SR latches to be reset whenever there is an "Ambiguity" ("No Symmetry Event") condition, or allowing the SR latches to ignore that situation and retain the last value. Numerous other approaches and derived signals may be realized, as will be demonstrated later in conjunction with Figure 16. The full functionality of a magnitude comparator is not needed to detect the symmetry events; for example, the circuit of Figure 7 could also be used.

7 Transition-Based Implementation

Next, some possible transition-based implementation approaches are considered. These effectively set states of a plurality of latching flip-flops responsive to the rising and falling

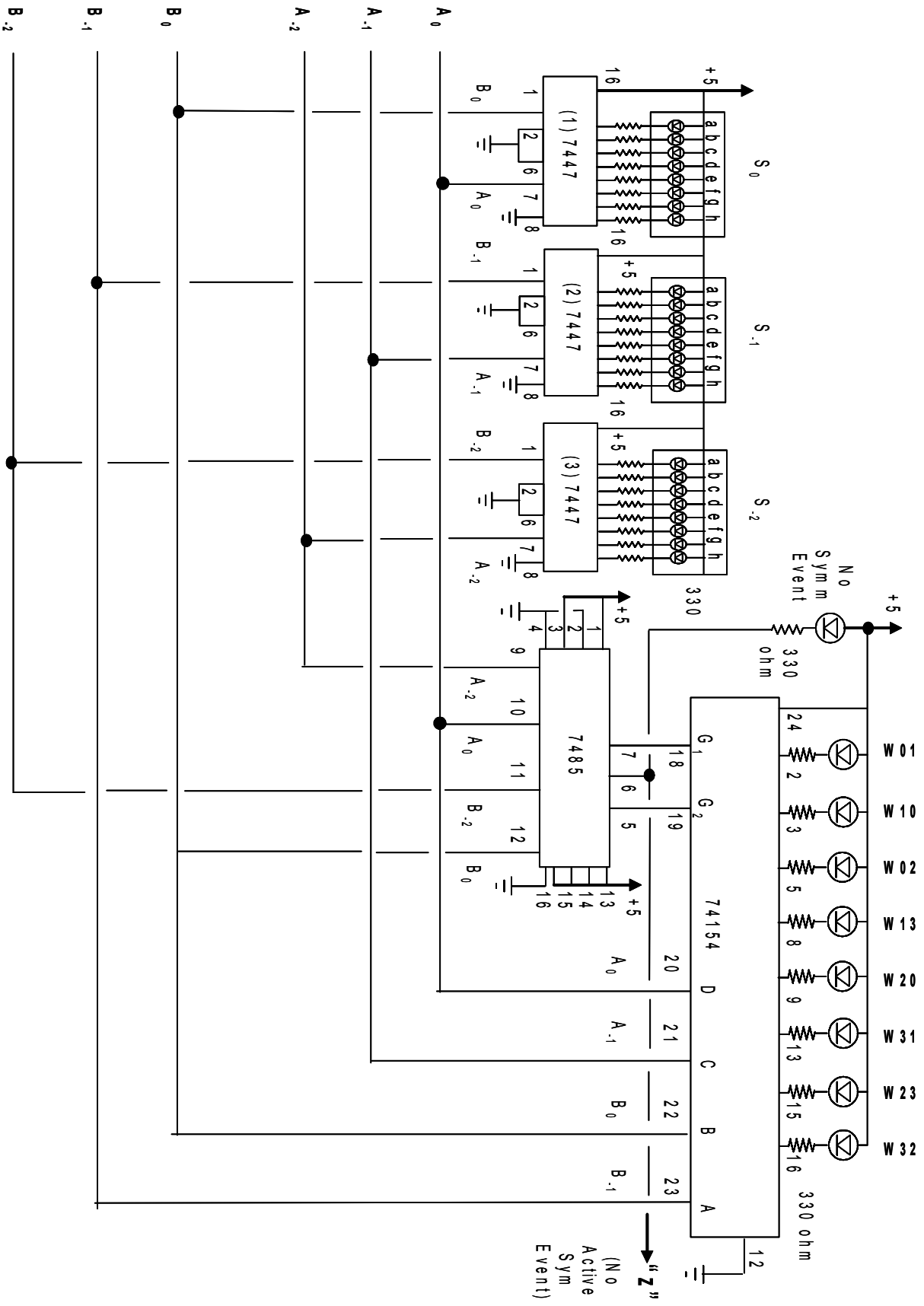


Figure 6c

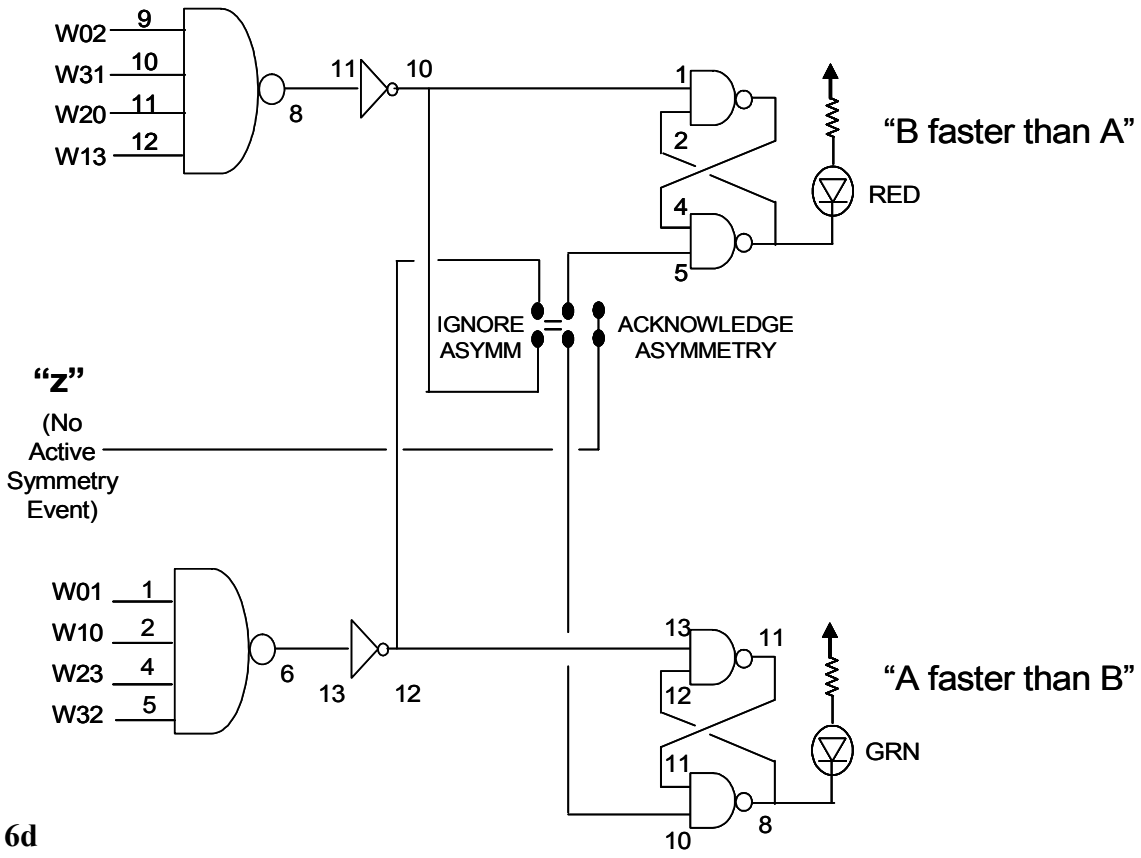


Figure 6d

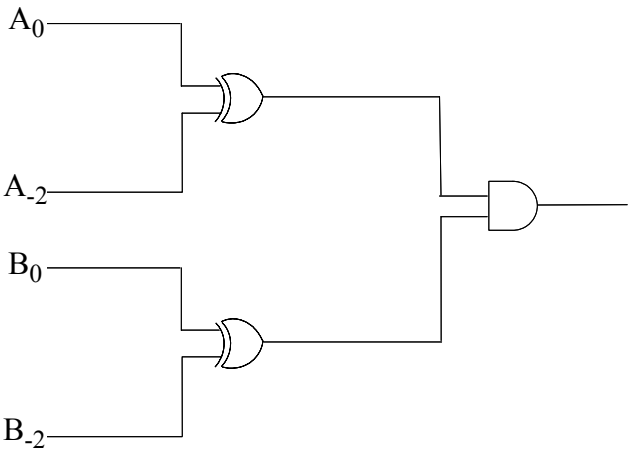


Figure 7

edges of the two square waves and apply combinational logic operations to the resulting state values.

As an orienting note, because transition-based implementations effectively set the state of a plurality of flip-flops with the rising and falling edges of signals and invoke combinational logic operations, the sample circuits may at first appear at a high level to resemble some types of existing edge-triggered frequency comparator circuits. However, these circuits are

completely different in principle, structure, and operation because they utilize only a feed-forward signal flow (i.e., there is no stored-state feedback) and require no quadrature signal inputs. These properties alone make the sample implementations to follow entirely different from existing edge-triggered frequency comparator circuits.

A transition-based implementation may identify the eight symmetry event conditions separately, as was done in the state-oriented implementation, or in related groupings ("equivalence-classes"). In state-oriented implementations, equivalence-classes are naturally implemented using "don't care" conditions across the grouping of states (utilizing the common practice of Karnaugh maps). Such detailed states need not be retained in transition-based implementations. It is possible to realize "don't care" structures across time, i.e., to detect classes of grossly similar phenomena independent of the fine-structure in the temporal ordering of events. However, this must be done carefully to avoid the situation depicted in Figure 2b.

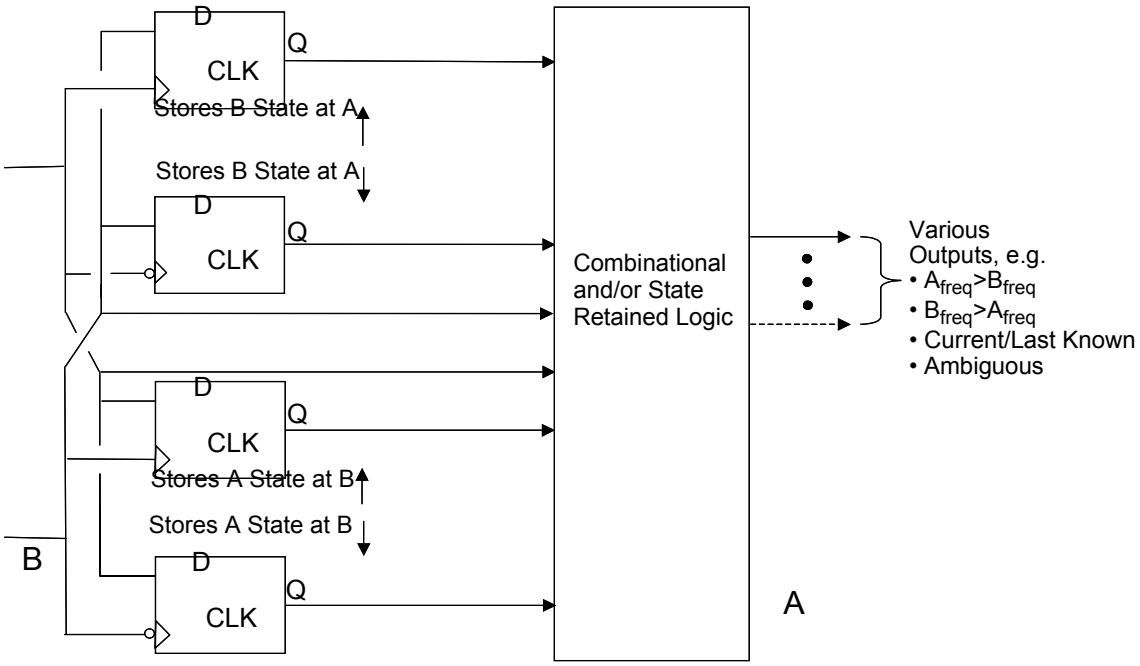


Figure 8

To illustrate the concern, Figure 8 depicts a problematic approach (one that ignores the third important remark made earlier). In this example, the rising and falling transitions of each square wave are used to trigger sampling of the value of the other square wave at that instant. The resulting data appears at first readily useful, but fails to be definitive. Referring to Figure 2b, the approach of Figure 8 will give the same results when the square wave B is compared to lower-frequency square wave A_1 or to higher-frequency square wave A_2.

The key condition in the sequence of transitions that must be captured is:

- the slower square wave must make a transition (either up or down);
- the faster square wave must make at least the next two transitions (either up then down or down then up);
- and only then may the slower square wave make its transition.

Data capture and pattern detection thus operate in a manner not unlike that of a combination lock that recognizes combination codes. As an additional caution, this approach to a transition-based implementation employs the use of both rising and falling edges of the square waves. Thus the circuit involves the co-presence of inversions and non-inversions of the same signal. The generation and handling of these inversions and non-inversions require care to prevent unnecessary limitations due to race conditions.

Figure 9a illustrates a first step in an abstract logic circuit realization of a transition-based implementation responsive to both rising and falling edges of symmetric square wave signals A and B that takes the above concerns into consideration. Here, four SR flip-flop latches are driven by transients of inverted and non-inverted versions of binary waveforms A and B. The notation Q(S,R) denotes the state of the latch output as a function of S and R:

S	**R**	**Q(**S,R**)**
0	0	Previous value of Q
1	0	1
0	1	0
1	1	1 (*pseudo-stable*)

Thus the configuration depicted in Figure 9a causes the four SR latches to behave as follows:

- The first latch output Q_1 is:
  - set to 1 when A makes a transition from 1 to 0;
  - set to 0 when B makes a transition from 1 to 0;
- The second latch output Q_2 is:
  - set to 1 when A makes a transition from 0 to 1;
  - set to 0 when B makes a transition from 1 to 0;
- The third latch output Q_3 is:
  - set to 1 when A makes a transition from 1 to 0;

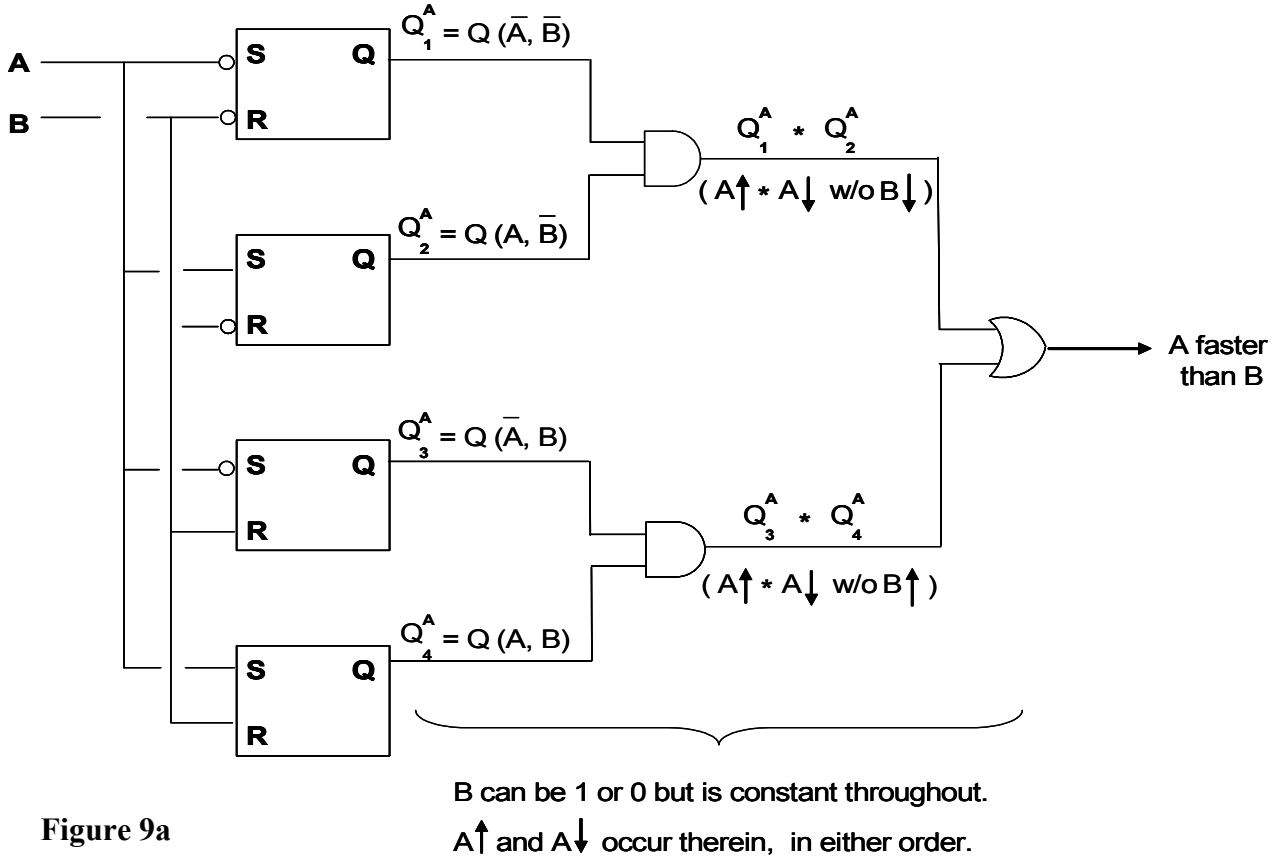


Figure 9a

- set to 0 when B makes a transition from 0 to 1;

- The fourth latch output Q_4 is:
  - set to 1 when A makes a transition from 0 to 1;
  - set to 0 when B makes a transition from 0 to 1.

If A makes both an up and a down transition (in either order) during an interval where B stays at 1 (i.e., makes no transition to 0), then outputs Q_1 and Q_2 are at some point both set to 1, and remain in that condition until B eventually makes a transition to 0 (which then causes both Q_1 and Q_2 to reset to 0). Referring to cases a and b of Figure 2c, these are two of the four cases in which the symmetric square oscillation of A must be faster than the symmetric square oscillation of B. An AND operation acting on outputs Q_1 and Q_2 produces a logical 1 under these first two "A faster than B" cases.

Similarly, if A makes both an up and a down transition (in either order) during an interval where B stays at 0 (i.e., makes no transition to 1), then outputs Q_3 and Q_4 are at some point both set to 1, and remain in that condition until B eventually makes a transition to 1 (which then causes both Q_3 and Q_4 to reset to 0). Referring to cases c and d of Figure 2c, these are the other two (besides a and b) of the four cases in which the symmetric square oscillation

of A must be faster than the symmetric square oscillation of B. An AND operation acting on outputs Q_3 and Q_4 produces a logical 1 under these second two "A faster than B" cases.

An OR operation acting on the AND outputs corresponding to the four "A faster than B" cases ultimately produces a logical 1 if A is faster than B.

By replicating the above arrangement with the roles of A and B reversed, one obtains the arrangement of Figure 9b, which ultimately produces a logical 1 if B is faster than A.

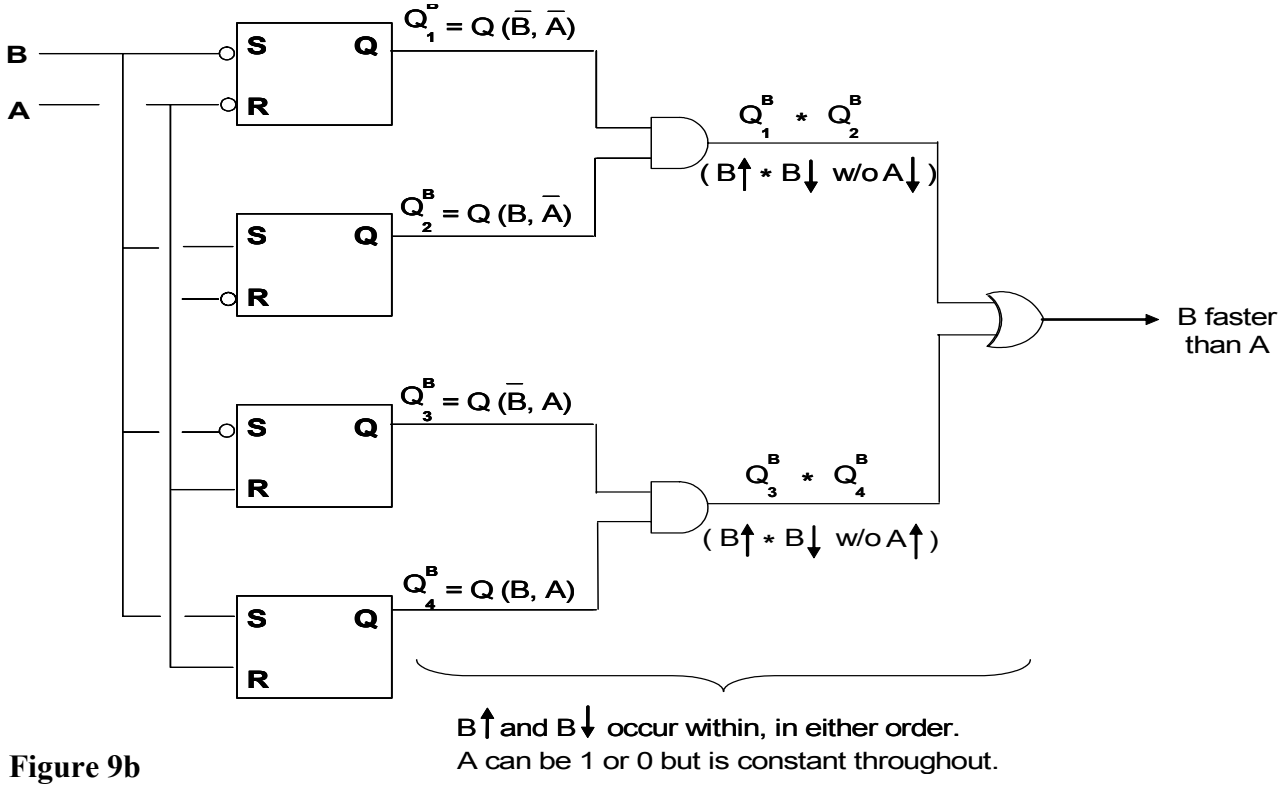


Figure 9b

Even though the configurations of Figures 9a-b use different latch driving arrangements, they are very similar. In fact, either may be converted to use the same latch driving arrangement as the other, allowing the two circuits to be readily merged into a form sharing the same four SR latches. This will be illustrated by converting the latch driving arrangement of Figure 9b into the latch driving arrangement of Figure 9a. Because of the symmetry of the SR latch, one has the inversion relation:

$$Q(S,R) = \bar{Q}(R,S)$$

Applying this to the values produced by the outputs of each of the four SR latches in Figure 9b yields the equivalent latch outputs depicted in Figure 9c. However, these complemented latch outputs are in fact readily provided by an SR latch (as shown later in Figure 10a) as the Q– output, thus immediately yielding the configuration of Figure 9d. Reversing inputs A and B now only causes the need to swap outputs of the second and third latches, resulting

in the configuration shown in Figure 9e. Comparing Figure 9e with Figure 9a shows the same latch driving arrangement, readily enabling the combined circuit of Figure 9f.

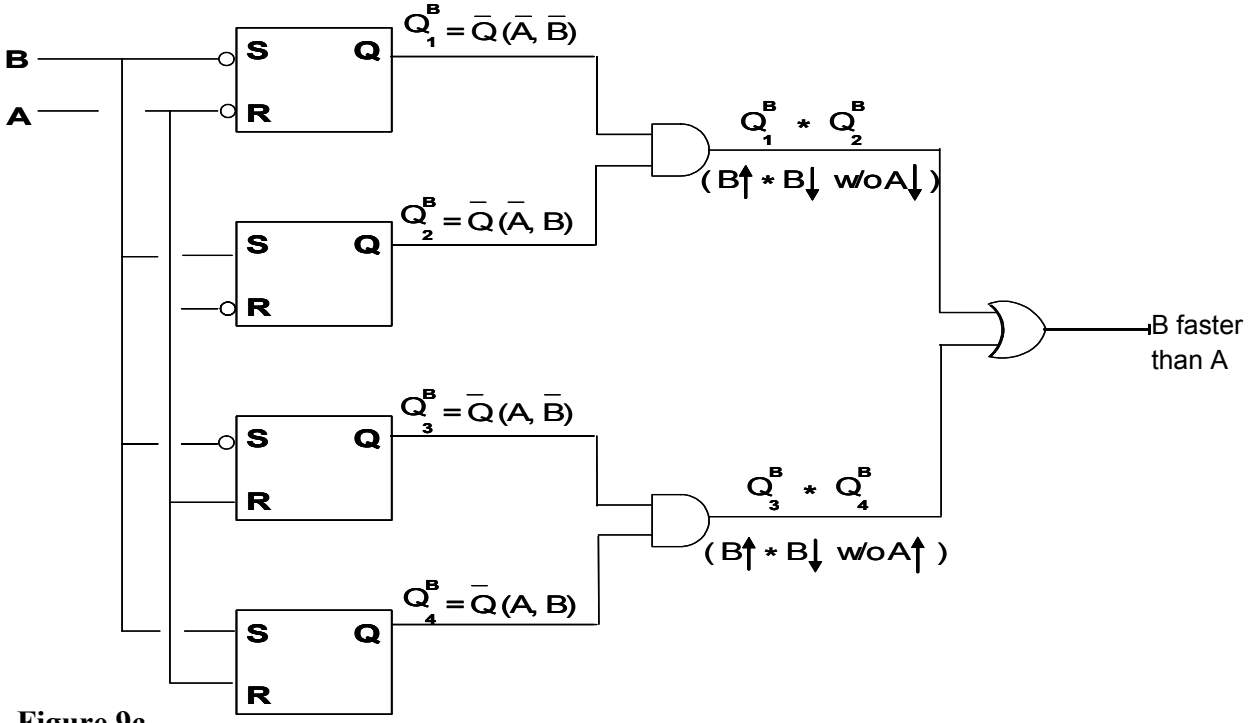


Figure 9c

Figure 10a shows a reference circuit realization of the arrangement of Figure 9f utilizing standard TTL/CMOS series integrated circuits. The pulse transition edges of applied signals A and B are isolated by 0.01*u*f capacitors driven by buffer and inverter circuits comprised of inverter elements. This transition-driven pulse generation arrangement is a simply-understood alternative to the circuits of Figures 4a-g, but has frequency-range limitations and can readily be replaced by circuits employing the techniques depicted in Figures 4a-g.

These four transition-driven pulse signals (W, X, Y, Z) are applied to inverted-input RS-latches formed from pairs of NAND gates. The inverted property of the latch inputs, together with the choice of capacitively coupled signals, are aligned to match the latch-driving arrangements of Figure 9f.

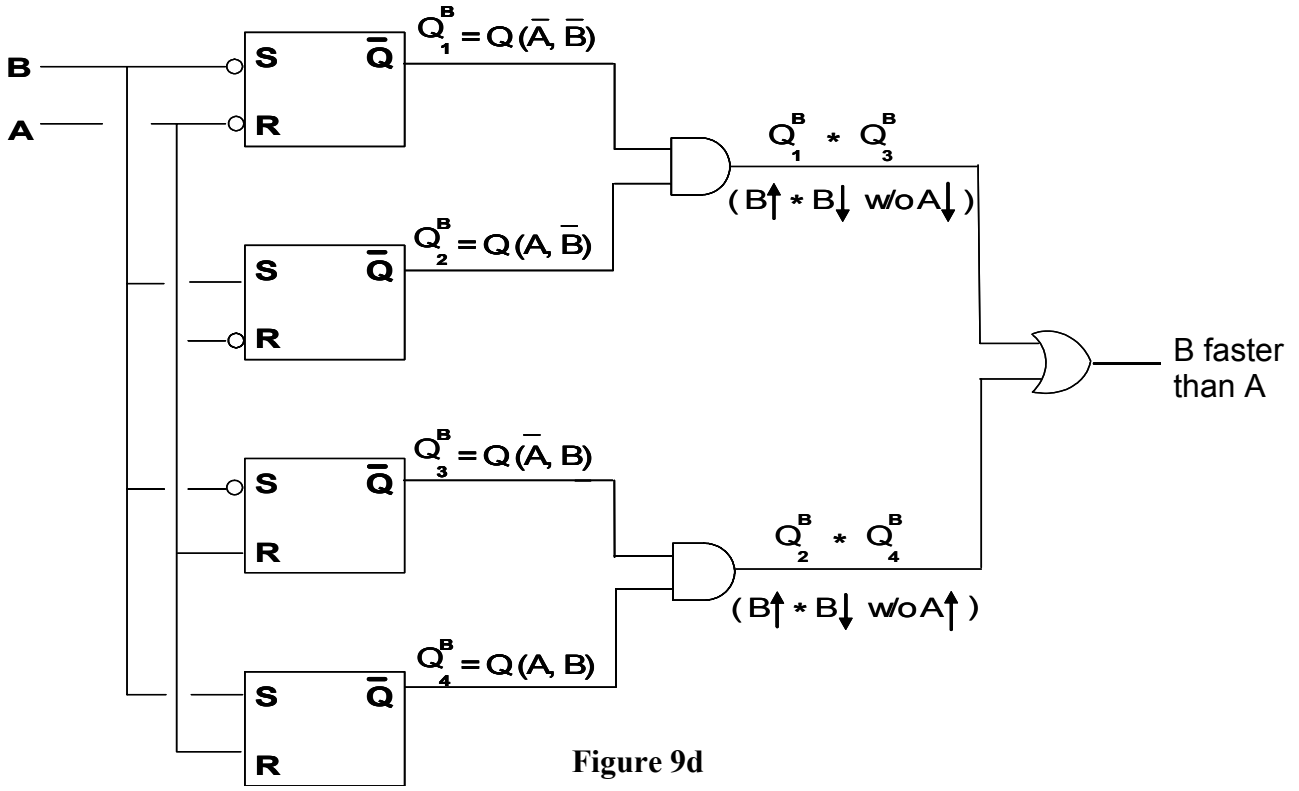


Figure 9d

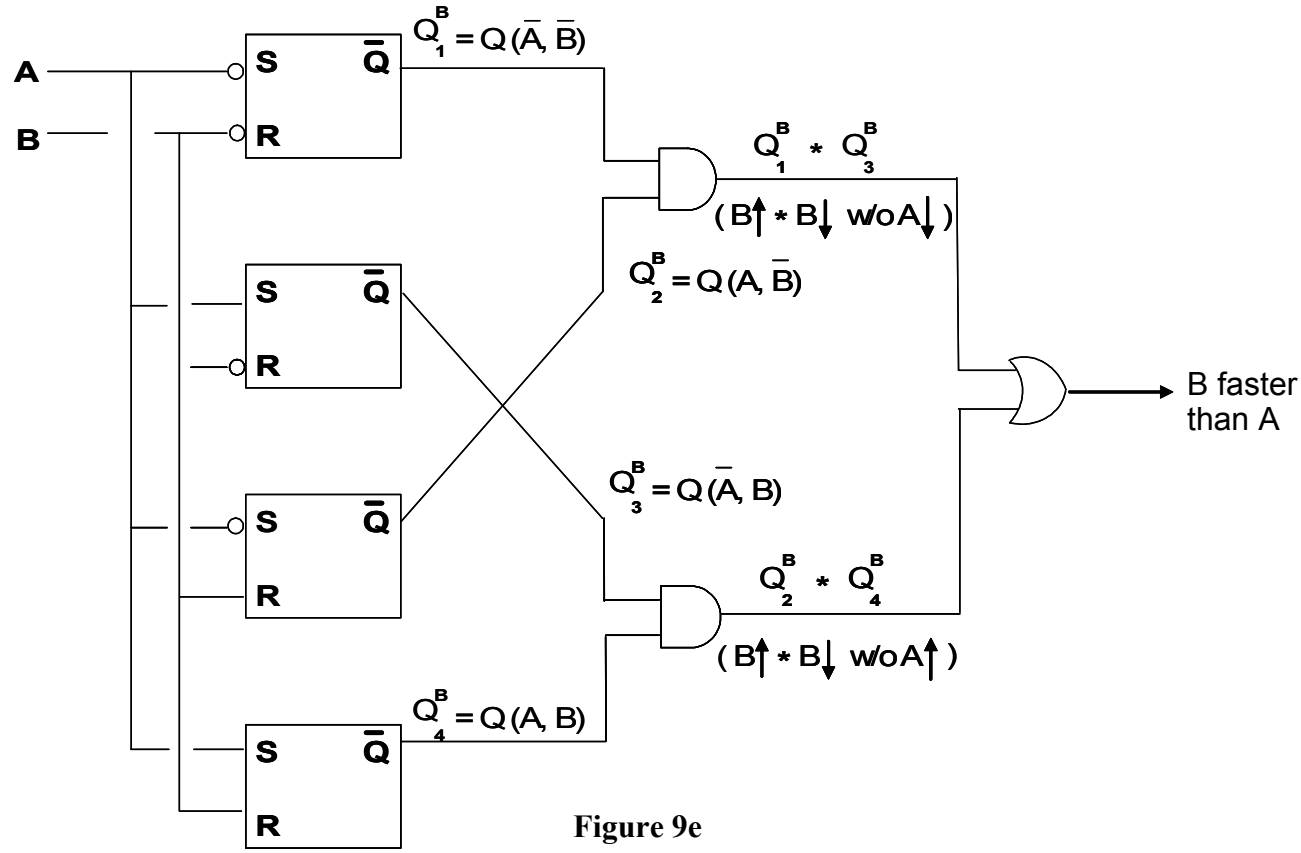


Figure 9e

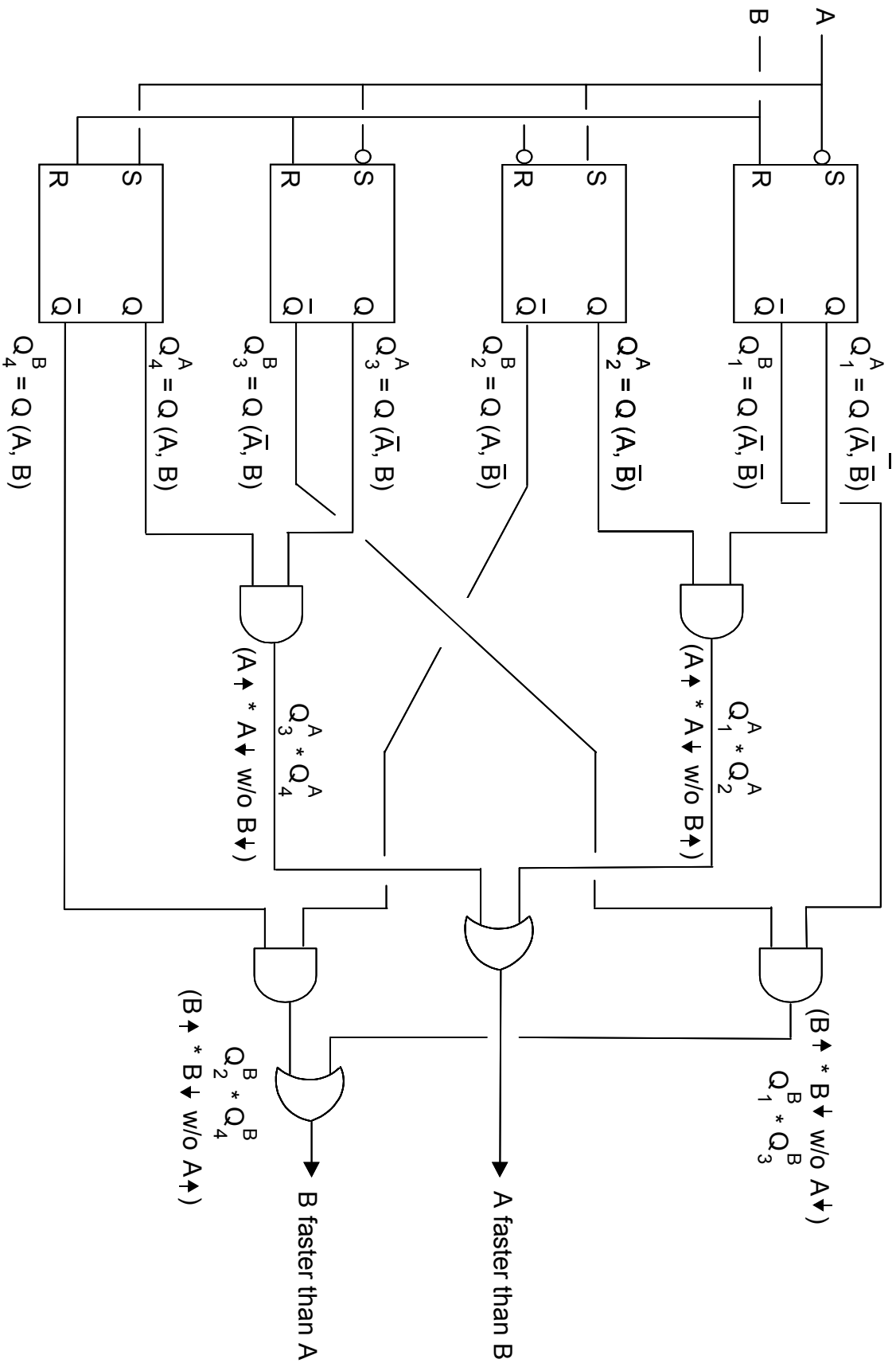


Figure 9f

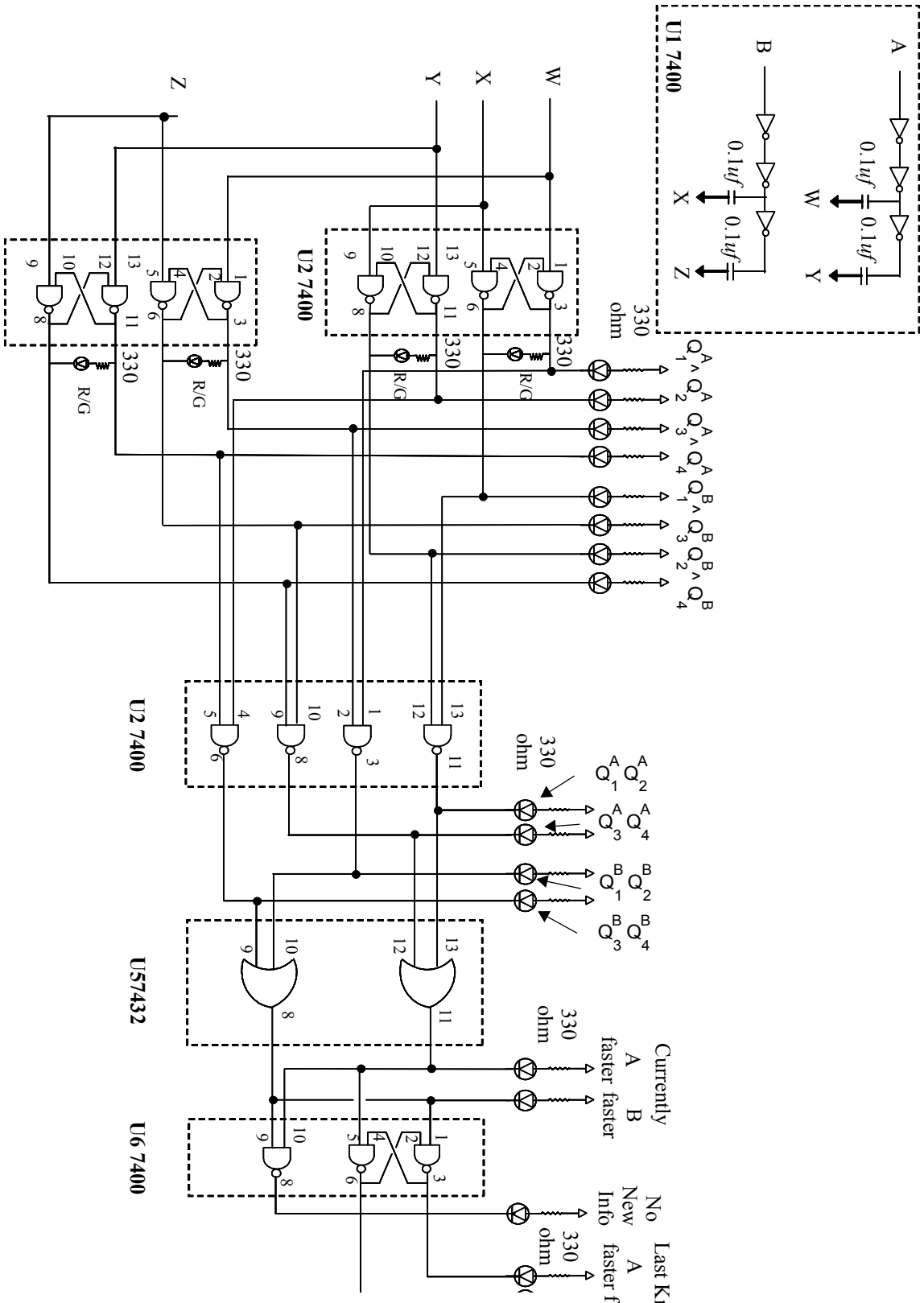

U1 7400
A
B
W
X
Y
Z
0.1uf
U2 7400
U2 7400
U2 7400
U57432
U6 7400
330 ohm
R/G
Currently A faster
Currently B faster
No New Info
Last Known A faster
Last Known B faster

Figure 10a

The four (NAND gate, inverted-input) RS latches provide both original and inverted versions of latch state output simultaneously, and these output signals are directed (in positive-valued logic) to four additional NAND gates. These latter four NAND gates produce (in inverted-valued logic) signals corresponding to the $Q_1^A \& Q_2^A$, $Q_3^A \& Q_4^A$, $Q_1^B \& Q_3^B$, $Q_2^B \& Q_4^B$ signals of Figures 9e and 9f. The inverted value of these logic signals is in keeping with inverted-value conventions of TTL/CMOS families of logic chips, should these signals be used for other purposes, and are also convenient for direct driving of indicator LEDs. These inverted-value logic signals are then applied to two OR gates, which by DeMorgan's law act as NAND gates on positive-valued logic signals. This gives inverted indications of "A faster than B" and "B faster than A," also in keeping with inverted-value conventions of TTL/CMOS families of logic chips and convenient for direct driving of indicator LEDs. When actual square waves are applied, the indications of "A faster than B" and "B faster than A" are constantly being reset when the slower waveform finally makes its transition. Thus by applying the inverted "A faster than B" and "B faster than A" signals to another NAND gate inverted-input RS latch, the last indication of which of input signals A and B had the higher frequency can be stored until that condition changes. Further, when no condition indicating that one or the other of input signals A and B had the higher frequency is currently active, there is no new conclusive information as to which of input signals A and B had the higher frequency. Under these circumstances, both of the outputs of the inverted "A faster than B" and "B faster than A" signals are high, so when they are applied to an additional NAND gate a signal indicating "No New Information Available" is produced, again in inverted-value form.

Figure 10b shows an LED demonstration adaptation of the reference circuit realization of Figure 10a. Here a number of LEDs are added to indicate the various logic levels of the signal flow, demonstrating operation. As indicated in the above discussion, the inverted-value signal conventions may be used for directly driving indicator LEDs tied to the power supply and fitted with an appropriate current limiting resistor (for example, 330 ohms for 5-volt TTL logic). Green LEDs are used to represent states of affairs relating to input signal A being faster than input signal B, and red LEDs are used to represent states of affairs relating to input signal B being faster than input signal A. In this convention, commonly available polarity-reversal bi-color LEDs (green for one current direction, red for the opposite current direction) may be attached, as shown across the complementary-valued outputs of the four NAND gate inverted-input RS latches. These, too, light green for states of affairs relating to input signal A being faster than input signal B, and red for states of affairs relating to input signal B being faster than input signal A. Finally, an additional LED, for example yellow, may be used to indicate cautionary "No New Information Available" status despite the final latched relative speed indication.

These LED indications readily verify the design theory when inputs A and B are driven with signals from an (electromechanically-debounced) pushbutton. When driven with low-frequency symmetric square wave oscillators, the LED indications visually demonstrate the patterns resulting from the various relative square wave conditions that lead to the frequency comparator output determination.

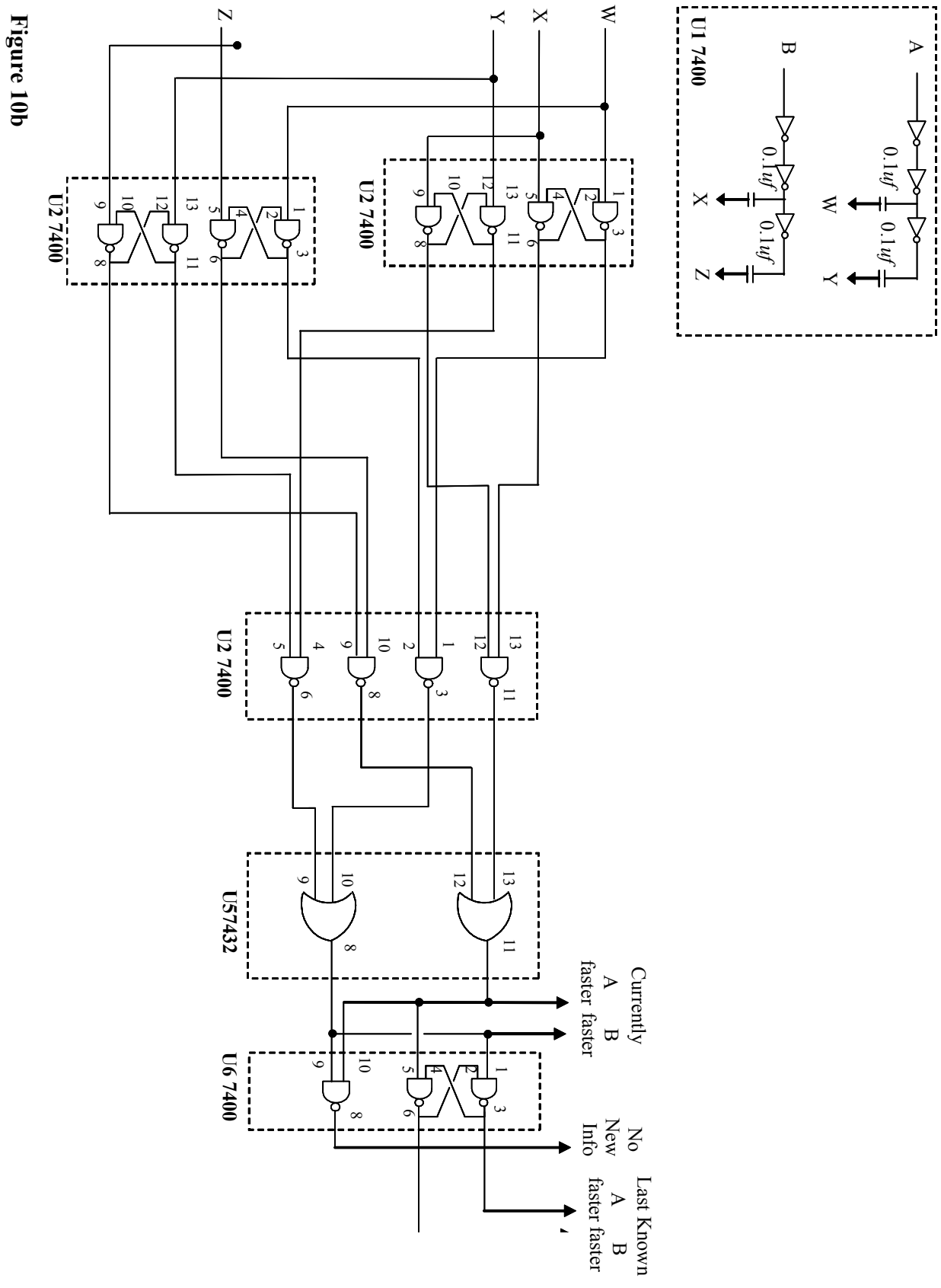


Figure 10b

When driven with signals from (electromechanically-debounced) pushbuttons, the circuit plainly demonstrates another application: it provides an indication as to which of two buttons or switches was last cycled between on and off with no change in the status of the other button or switch. This observation provides further motivation for a chip implementation of at least the transition-based embodiment of the invention, as it can be used not only as a frequency comparator but also in user-interface, sequence-auditing, and other industrial applications.

8 Signal Source for Demonstration of the Symbol-Based and Transition-Based Embodiments

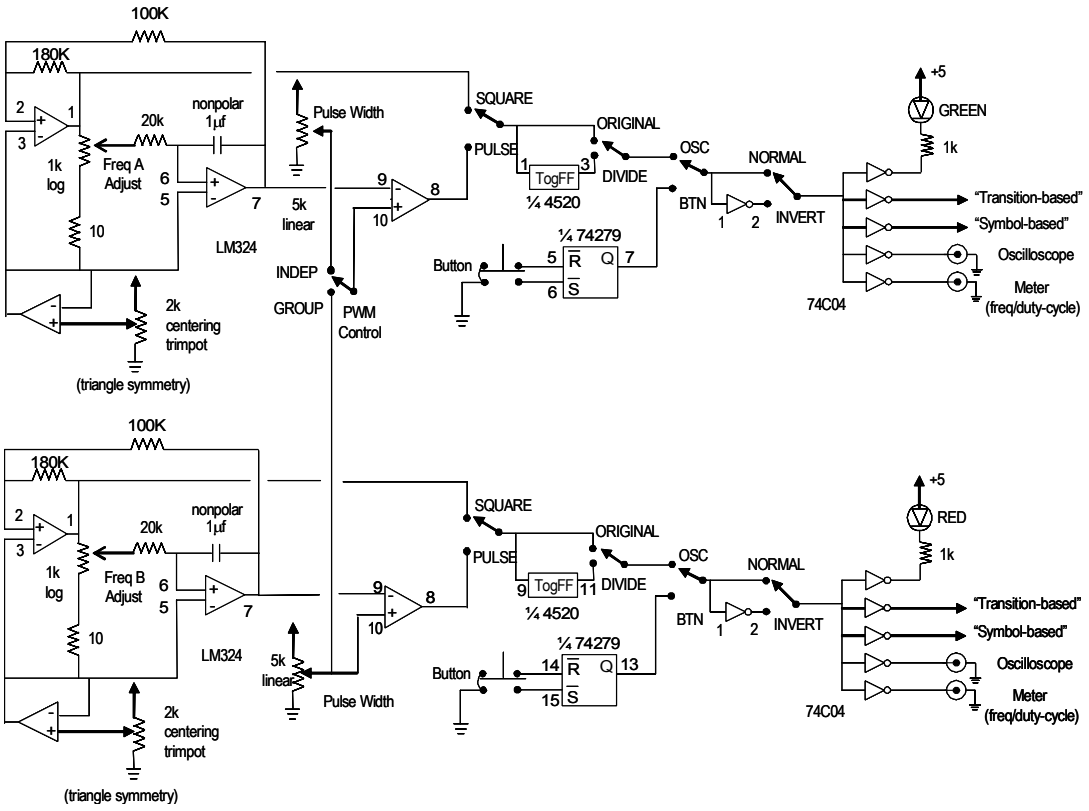


Figure 11a

Figure 11 shows a demonstration signal source providing pushbutton actuation and variable frequency low-frequency oscillators with controllable symmetry useful in exercising the configurations of Figures 10b and 6a-c. This signal source comprises two nearly identical independent versions of the same circuit. The only exception is a switch that allows for pulse widths to be set and adjusted separately (in which case the two circuits indeed are fully identical and independent) or set with a common pulse width adjustment control. Each

nearly identical circuit comprises an adjustable low-frequency oscillator, which comprises an integrator and a comparator in a positive feedback loop. The comparator output is a binary-valued periodic waveform very closely resembling (if not matching) that of a symmetric square wave, while the integrator output is a continuous-valued periodic waveform very closely resembling (if not matching) that of a symmetric triangle-wave. Such a low-frequency oscillator is well known in the art of analog music synthesizers. The comparator compares the integrator triangle wave output to a fixed reference voltage, set to a value half that of the amplitude of the triangle wave.

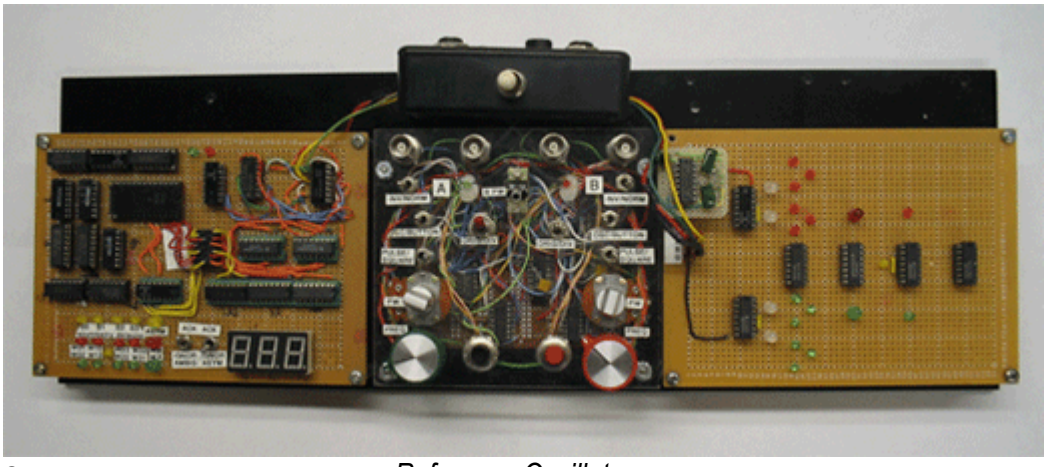
The circuit of Figure 11a provides this reference voltage by buffering the output of a trimpot that may be adjusted for maximal symmetry of the overall output (or for precise duty-cycle matching when the two halves of the circuit share the same pulse-width adjustment control, to be described next). The output of the comparator is variably attenuated by a logarithmic potentiometer panel control arrangement to provide variable frequency control. A small resistor (here 10 ohms) between the logarithmic potentiometer panel control and the buffered reference voltage sets the lower limit of the frequency range possible with the other components and configuration involved.

The triangle wave is additionally directed to a second comparator that compares the integrator triangle wave output to a freely adjustable voltage set by a linear potentiometer panel control. This allows variation in the duty-cycle of the resultant pulse waveform. A switch allows, as described earlier, pulse widths of the two oscillators to be adjusted separately with individual panel controls (“INDEPENDENT”) or together (“GROUP”). In the second case, oscillator A and oscillator B share the same pulse width adjustment panel control.

A switch is provided for each oscillator to permit selection of a nearly-symmetric square wave or a controllably asymmetric pulse waveform. The result is passed to a toggle flip-flop that restores symmetry under all conditions while also dividing the frequency by a factor of 2. An additional switch is provided for each oscillator to permit selection of the undivided (“ORIGINAL”) waveform or the symmetric frequency-divided (“DIVIDED”) version. The resulting selection is then passed to yet another switch provided for each channel, which permits selection of this signal or that of a debounced pushbutton. The pushbutton is debounced using an RS latch. The resulting choice is then passed to an inverter and an additional switch provided for each channel configured to select between the original (“NORMAL”) and inverted (“INVERT”) versions of the signal created thus far. At this point the signal is directed to one or more additional buffering stages for driving external circuitry. Here inverters are used to drive an LED and to provide separate outputs to the symbol-based and transition-based embodiments (such as those of Figures 6a-c and 10a-b), as well as to test instruments such as oscilloscopes, phase meters, and frequency meters. Other implementations may add additional inverters to change the logic sense of the signal outputs, employ op amps for better current drive and static electricity immunity, etc. As to phase and frequency measurement, it is noted that the quite inexpensive Extech model MN26 DVM provides adequately precise phase and frequency measurements for use with the demonstration circuits of Figures 6a-c and 10a.

The signal source just described facilitates relatively deep introductory study of the properties of pairs of asynchronous square waves and of the present invention as described thus far. It also provides a way to explore the behaviors of pairs of asynchronous asymmetric binary waveforms, to be discussed in a subsequent section.

Figure 11b illustrates a prototyping, demonstration, and research setup employing this flexible signal source and realizations of the state-based and transition-based circuits described above.



State-Based Implementation (with asymmetry handling) *Reference Oscillators (with sym/asym features)* *Transition-Based Implementation*

9 Single-Chip System and SoIC Implementations

The symbol-based or transition-based embodiments may be implemented as a subsystem of discrete components; as a devoted functional integrated circuit; as a configuration of standard cells in a gate array, FPLA, ASIC, VLSI on a chip;, or as an "IP core" within a System-on-a-Chip ("SoIC).

The design of Figure 6a can be readily adapted to an integrated circuit implementation via appropriate transformations and simplifications. For example, the magnitude comparator chip reduces the wiring and chip count in the Figure 6a implementation, but has far more functionality than is required to detect symmetry-events; it may be replaced by a more focused symmetry-event detector such as that illustrated in Figure 7. Similarly, the 4-bit demultiplexer, convenient for showcasing the individual symmetry-event detections, can be replaced with more focused combinational logic, potentially leveraging as opportune the extensive structures and symmetries identified earlier. Additionally, the shift-registers may be restructured as alternative flip-flop configurations. Further, the entire functionality, or variations of it, may be adapted into an algorithm, as will be described later in conjunction with Figures 13a-b.

Similarly, the arrangement of Figure 9f and its reference circuit realization of Figure 10a may be even more readily adapted to integrated circuit implementation. As described earlier, a chip implementation of at least the transition-based embodiment of the invention can be used not only as a frequency comparator but also in user-interface, sequence-auditing, and other industrial applications.

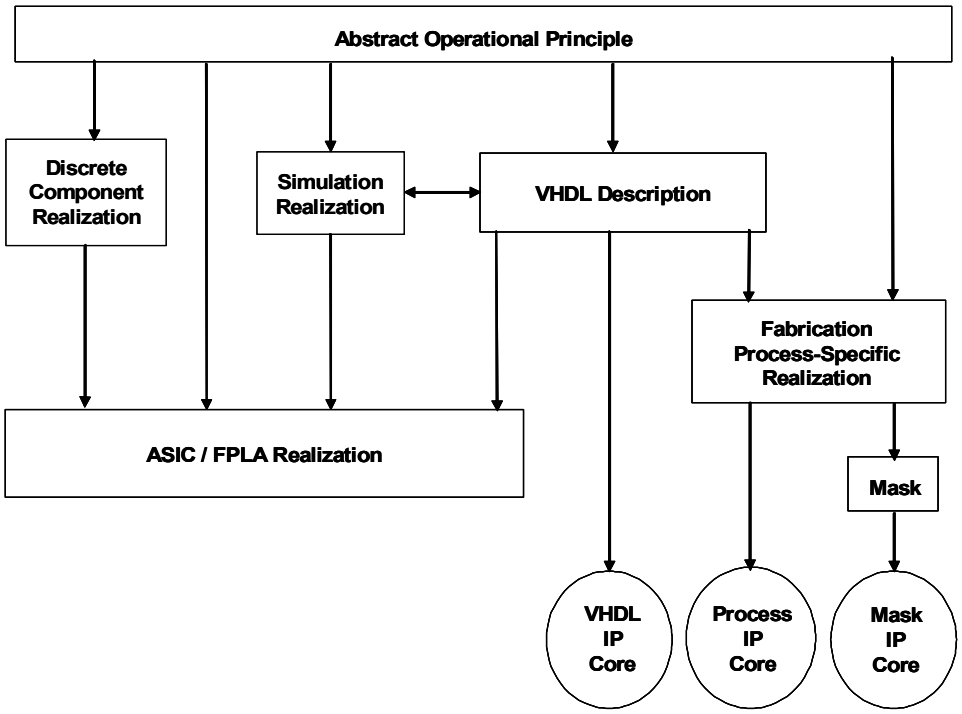


Figure 12a

Figure 12a illustrates a wide range of silicon-based embodiment options for the invention. A chosen abstract operating principle provided for by the invention may be realized directly employing standard TTL/CMOS components in ways similar to those described in conjunction with Figures 6a-c and Figures 10a-b. Alternatively, the same abstract operating principle provided for by the invention may be realized utilizing gate and/or cell primitives inherent to specific ASIC, FPLA, and related technologies. Additionally, the abstract operating principle provided for by the invention may be realized utilizing primitives inherent to a specific fabrication process. In a more measured approach, the abstract operating principle provided for by the invention may be encoded into a simulation-oriented realization. In some simulation environments, the simulation description may be directly applied to specific ASIC, FPLA, and related technologies. Other types of simulation environments work with Very-High-level Description Language (VHDL) systems that in turn can be directly applied to specific ASIC, FPLA, and related technologies or a specific fabrication process.

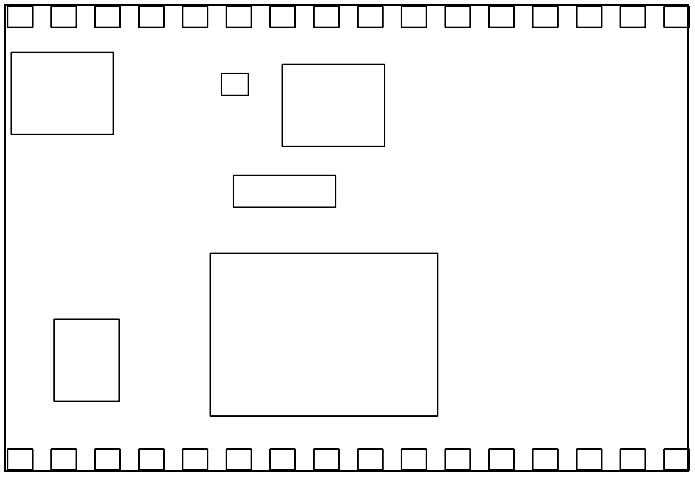


Figure 12b

Figure 12a also illustrates a number of ways the aforementioned realizations may be adapted to create an IP core for use as a subsystem in a larger functional integrated circuit, gate array, FPLA, ASIC, VLSI, or SoIC. VHDL environments may be packaged to create a VHDL-defined IP core. Alternatively, fabrication process-specific realizations may be packaged at a high-level to create a process-specified IP core, or packaged as a subsystem mask to create a mask-specified IP core. Figure 12b illustrates a number of subsystems that could make up a larger system in a large functional integrated circuit, gate array, FPLA, ASIC, VLSI, or SoIC, in which an implementation of the present invention may comprise one of the subsystems.

10 Algorithmic Embodiments

The invention provides for algorithmic implementation of the symbol-based, transition-based embodiments presented earlier.

As an example, Figure 13a shows a flowchart of an algorithmic embodiment of a symbol-based implementation operating in a time-driven fashion as depicted in Figure 1f. The ongoing dataflow begins with a comparison of the current values of input signals A and B. If these values can be quantized into binary values, the resultant pair of binary values (one for each input signal) is then converted to an appropriate "current symbol" $\{S_0, S_1, S_2, S_3\}$ as described earlier and summarized in Figure 2c. This current symbol is compared to the stored "previous symbol" to determine if the current symbol has changed since it was last stored. If it has not, no action need be taken (although the current symbol may overwrite the identical previous symbol, should that prove useful in an implementation). If the symbol has changed, the previous symbol is stored as the new "legacy symbol," and the current symbol is stored as the new previous symbol. The current symbol and the legacy symbol are compared, and if they are identical a symmetry event has occurred. The particular symmetry event is determined by the current symbol and the previous symbol. These may be used either to directly determine which signal has the higher frequency, or to compute a

corresponding symmetry event "word" $\{w_{01}, w_{02}, w_{10}, w_{13}, w_{20}, w_{23}, w_{31}, w_{32}\}$, which is subsequently interpreted to determine which signal has the higher frequency.

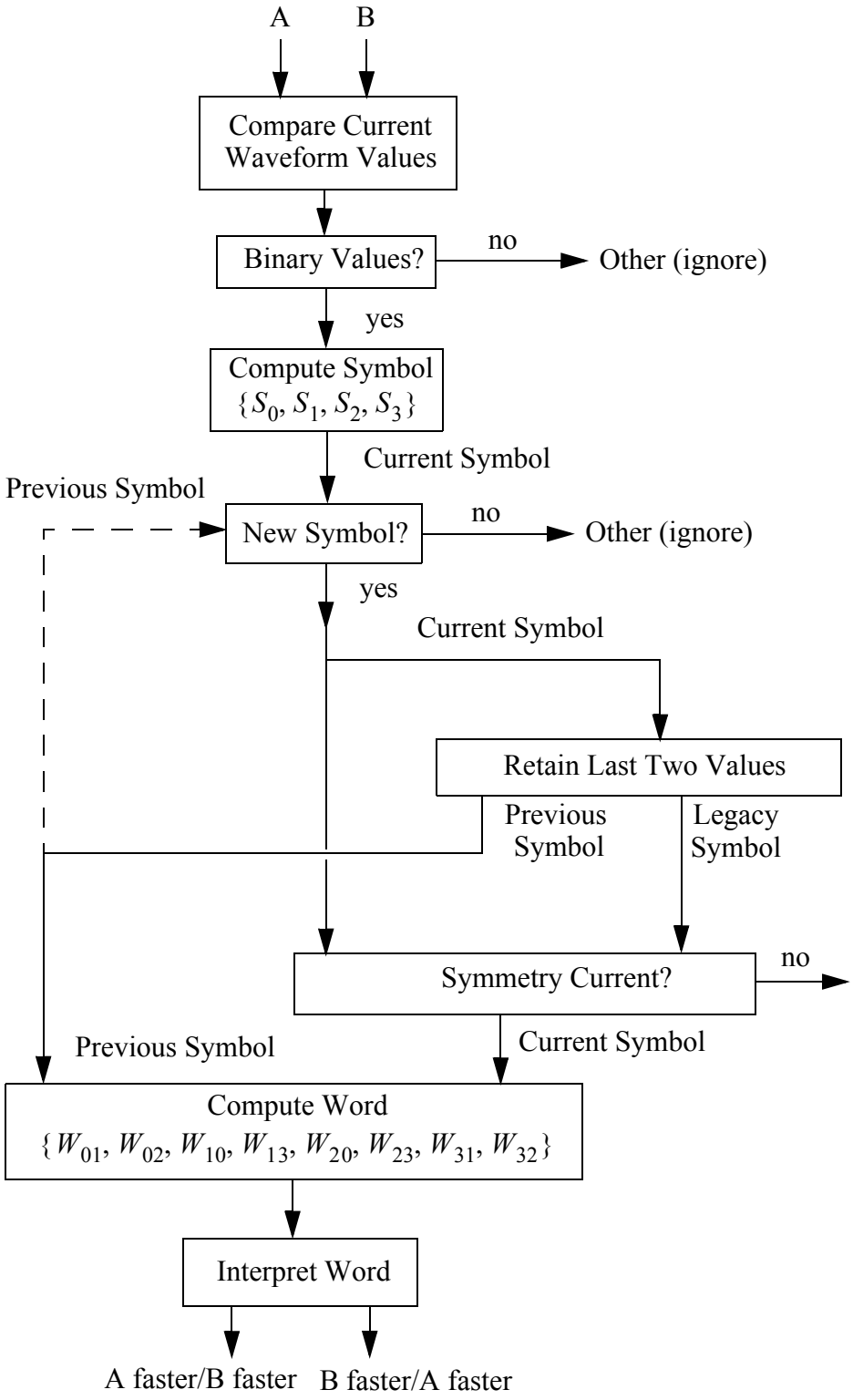


Figure 13a

An algorithmic embodiment of a transition-based implementation operating in a time-driven fashion, as depicted in Figure 1f, follows directly from the operational sequence depicted in Figures 9f and 10a-b. A flow chart is provided in Figure 13b. Referring to the figure,

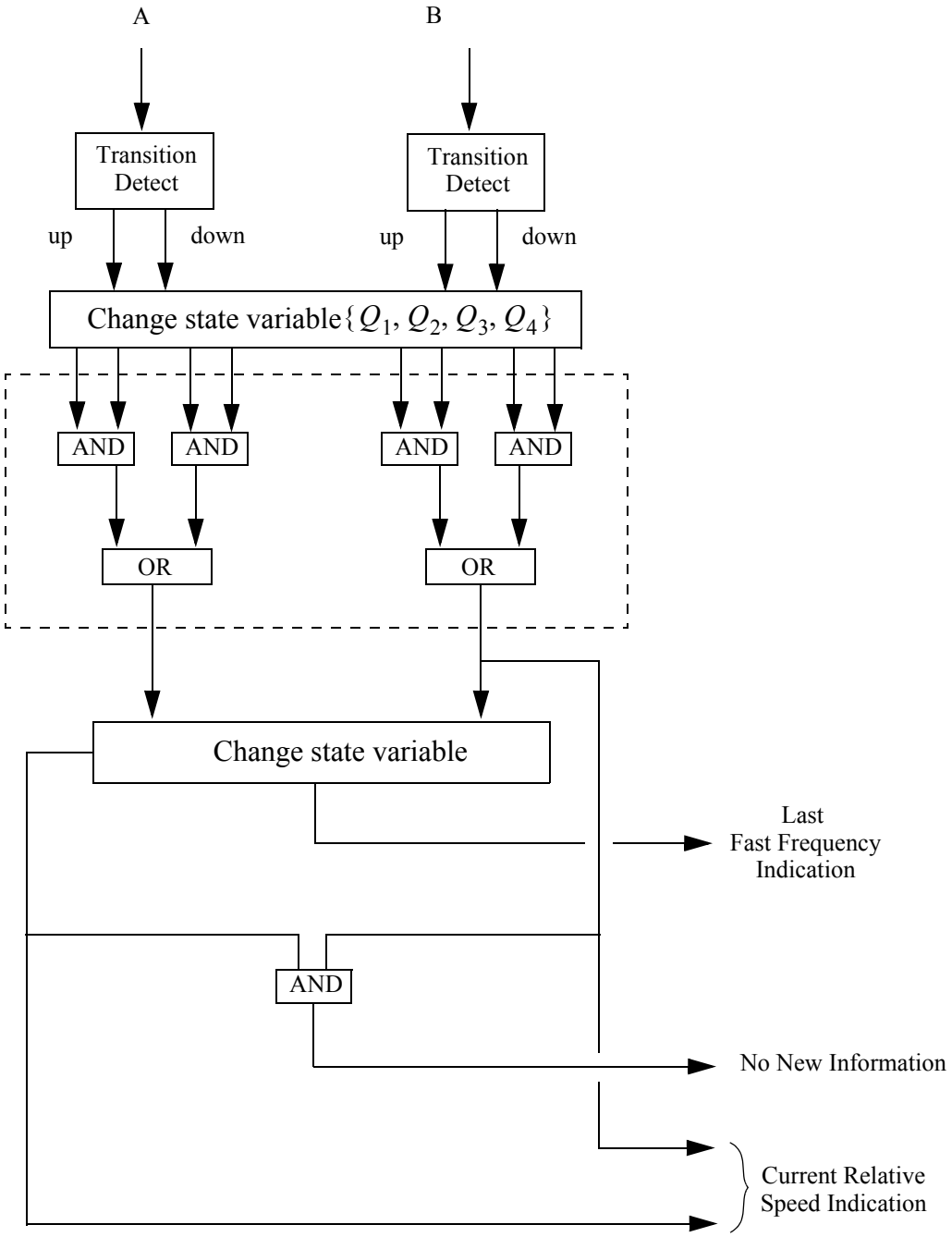


Figure 13b

binary-valued input signals A and B are checked for transitions, and any detected transitions are used to change the state of four stored binary variables Q_1, Q_2, Q_3, and Q_4 in a fashion consistent with Figures 9f and 10. These variables are acted on by four AND operations, collectively producing results corresponding to the $Q_1^A\&Q_2^A$, $Q_3^A\&Q_4^A$,

Q_1^B&Q_3^B, Q_2^B&Q_4^B signals of Figure 9f. These results are then applied to two OR gates, producing pulsed indications of "A faster than B" and "B faster than A" conditions. These indications are then applied to set the state of a variable whose state denotes the last indication of which of input signals A and B had the higher frequency. The pulsed indications are also directed to an additional AND operation, producing a result indicating that "No New Information" is available.

11 Extension to Three or More Input Signals

The invention provides for extensions to handle three or more input waveforms. This may be done in a number of ways. In one approach, the input waveforms are compared in all possible pairs. The outcomes are then directed to additional logic operations or computations to identify which waveform is fastest, and, potentially, additional ordering information. As an example, consider the case of four input signals A, B, C, and D. Comparing four signals two at a time in all combinations requires (by standard combinational formulas) six comparisons. If the pair-wise comparison outcomes are as follows:

- (Frequency of C) > (Frequency of B)
- (Frequency of C) > (Frequency of A)
- (Frequency of C) > (Frequency of D)
- (Frequency of B) > (Frequency of A)
- (Frequency of B) > (Frequency of D)
- (Frequency of A) > (Frequency of D)

then the overall uniquely determined result is:

(Frequency of C) > (Frequency of B) > (Frequency of A) > (Frequency of D)

In this first ("all pair-wise combinations") approach, for N input signals the number of required frequency comparator stages grows as N!/[(N-2)!2!] with increasing number of inputs N. In a second ("cascading") approach, the N input signals are separated into N/2 groups. If N is even, the N/2 groups will comprise N/2 pairs. If N is odd, the N/2 groups will comprise (N-1)/2 pairs plus one additional signal. Each of the input signal pairs is independently compared to choose the faster of each pair. The faster of each pair is passed to a subsequent stage where this process is repeated. A first stage will thus comprise on the order of N/2 frequency comparators, a second stage will comprise on the order of (N/2)/2 = N/4 frequency comparators, a third stage will comprise on the order of (N/4)/2 = N/8 frequency comparators, etc. In this approach, for N input signals the number of required fre-

quency comparator stages grows roughly as $N\log_2 N$. For N>6, the second approach is of less complexity, particularly if N is reasonably large. For N=20 the second approach has about half the complexity of the first approach, for N=40 the improvement is by a factor of approximately three, and for N=100 the improvement is by a factor greater than seven.

Figure 14a illustrates an example of a circuit that propagates a selected input from a pair of inputs based on the outcome of a frequency comparator. When the pair of applied inputs is the same as applied to the frequency comparator, the circuit will propagate the input signal of faster frequency (or slower frequency, depending upon the connections and logic value assignments). Figure 14b shows this arrangement being used with three frequency comparators to identify the fastest of four input signals. The results shown, provide the faster of each pair of comparisons and a logical indication as to which of these is faster. These may be used directly if useful in that form or be directed to a subsequent instance of the circuit of Figure 14a to deliver the fastest (or in a complementary implementation, slowest) input signal. Alternatively, the indication as to which input of each pair is faster may be subjected to logical processing along with the output of the last frequency comparator to identify which of the four signals is faster. Figure 14c shows a similar arrangement for the case of three input signals.

In this fashion, one or more instances of the circuits of Figures 14a-c may be ganged in various combinations to realize a circuit implementation of the second cascading approach.

12 Extension of the Approach for Handling Asymmetric Pulse Waves

Many common oscillators constructed from feedback loops introduced around logic gates (for example, CMOS inverter gates) produce slightly asymmetric waveforms that differ in duty-cycle from that of complete symmetry (50%) by values such as 3% or more. In practice, all square wave oscillators will produce binary-valued waveforms that are at least slightly asymmetric.

Additionally, when the frequency of a square wave oscillator is modulated, asymmetries in pulse width are introduced as the waveform period expands or contracts. If the frequency is modulated rapidly with respect to the oscillator frequency or with a large modulation index, these asymmetries can be significant. If the frequency is modulated relatively slowly with respect to the oscillator frequency or with a small modulation index, the asymmetries are insignificant.

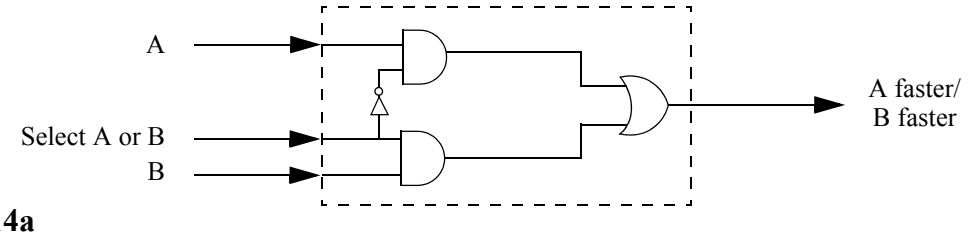


Figure 14a

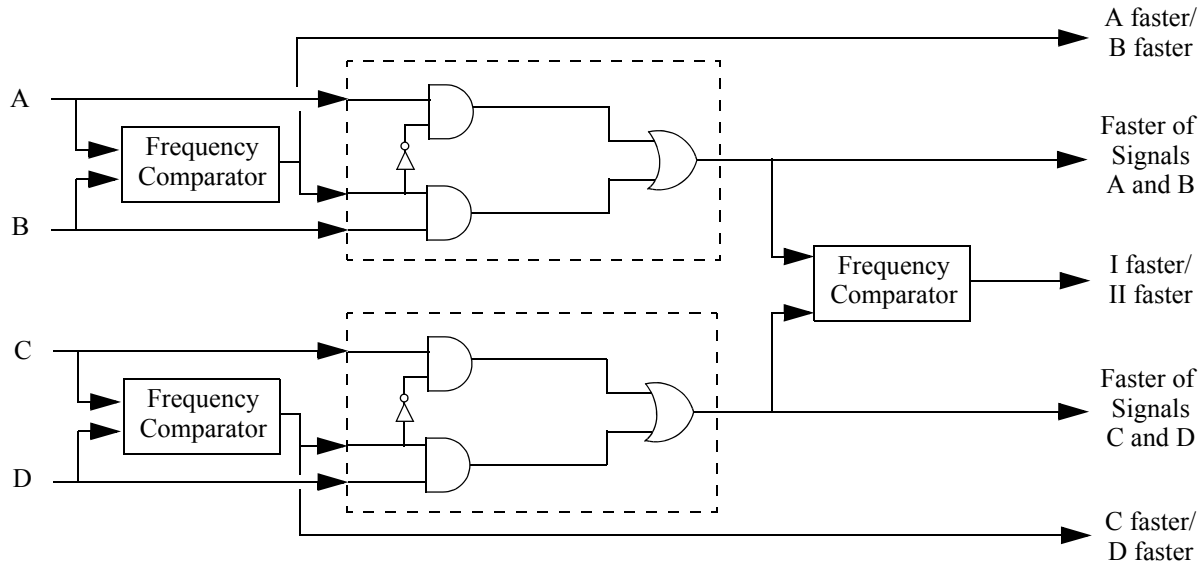


Figure 14b

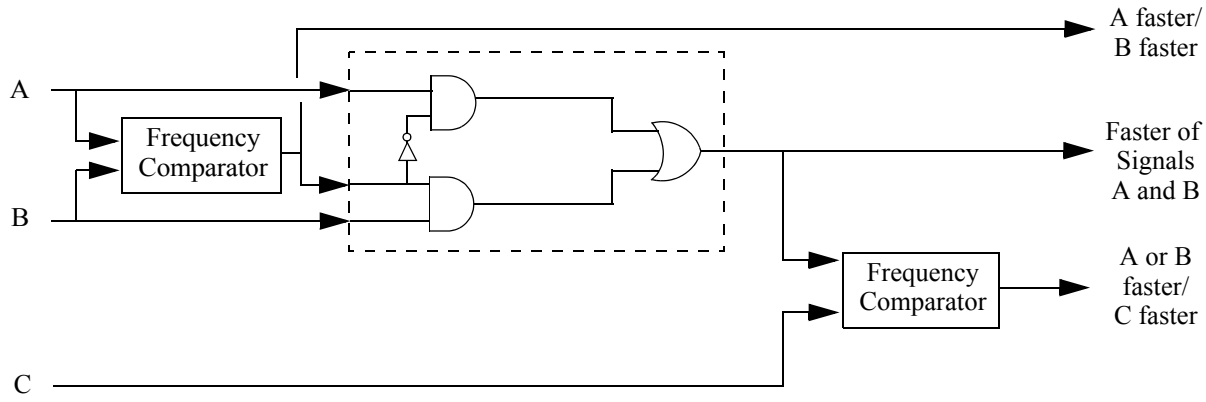


Figure 14c

When an asymmetric binary-valued pulse waveform is compared to another waveform whose frequency is sufficiently close, a number of additional phenomena can occur:

- the narrower portion of one asymmetric square wave experiences an enveloping event with respect to either:

- symmetric portions of a fully symmetric ideal square wave, or
- the wider portion of the other asymmetric square wave.
- the wider portion of one asymmetric square wave experiences an enveloping event with respect to either:
- symmetric portions of a fully symmetric ideal square wave, or
- the narrower portion of the other asymmetric square wave.

These additional phenomena can confuse the methods described thus far.

The degree to which these phenomena can occur is bounded by how close the waveforms are in frequency and how asymmetric each of the waveforms is. In cases where the frequencies of two compared waveforms are close enough to create these phenomena, a natural solution for many applications is to employ toggle flip-flops to homogenize the symmetry as described earlier in conjunction with Figure 3c. However, various types of asymmetric conditions and the phenomena they induce can be readily detected, and in many cases adverse effects may be readily corrected. This section considers extensions of the invention to handle asymmetric binary-valued pulse waveforms.

When the input to implementations of the invention designed specifically for symmetric waveforms are asymmetric waveforms, and when the frequencies of the asymmetric waveforms are sufficiently close, alternating indications that each of the waveforms is faster than the other can result. Referring to Figure 15a-d, four cases (polarity combinations) of asymmetric pulse waves of almost the same frequency and almost the same (asymmetric) duty cycle are shown. As may be seen in this figure, under these conditions asymmetric pulse waveforms have unique signature events as well, expressible in terms of "signature" sequences of symmetry events:

- if the enveloping event resulting from the asymmetry involves brief flashes of the S_0 symbol, there will be a sequence of w_{01} and w_{02} symmetry events separated by one non-symmetry event (Figure 15a);
- if the enveloping event resulting from the asymmetry involves brief flashes of the S_1 symbol, there will be a sequence of w_{10} and w_{13} symmetry events separated by one non-symmetry event (Figure 15b);
- if the enveloping event resulting from the asymmetry involves brief flashes of the S_2 symbol, there will be a sequence of w_{20} and w_{23} symmetry events separated by one non-symmetry event (Figure 15c);
- if the enveloping event resulting from the asymmetry involves brief flashes of the S_3 symbol, there will be a sequence of w_{31} and w_{32} symmetry events separated by one non-symmetry event (Figure 15d).

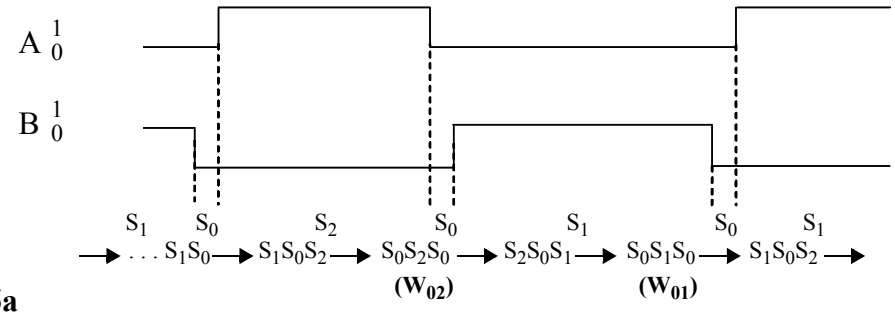


Figure 15a

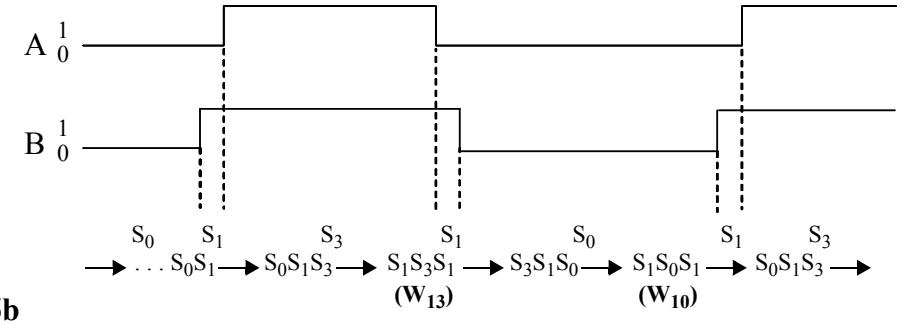


Figure 15b

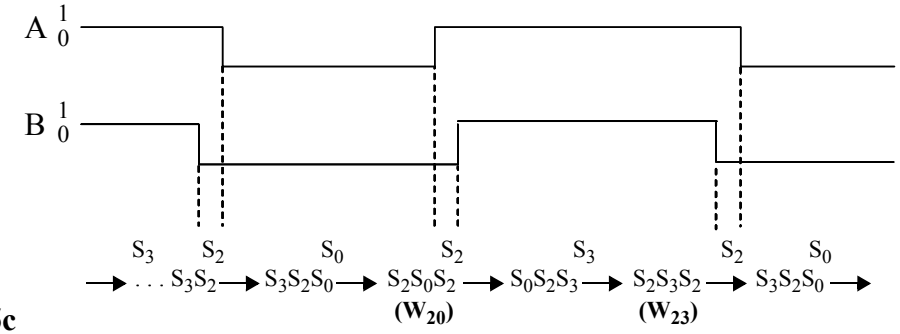


Figure 15c

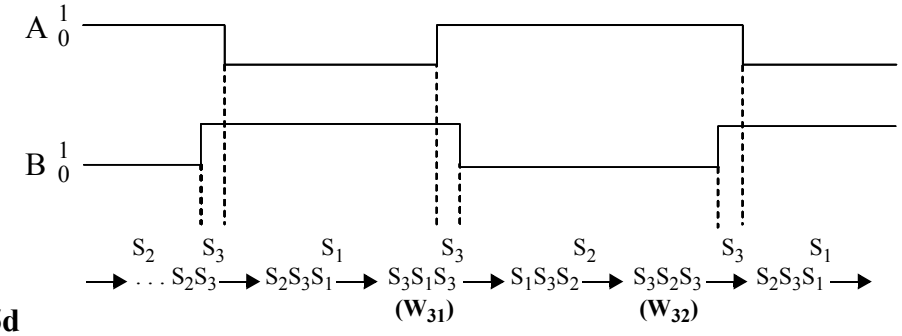


Figure 15d

Note that the pairs of symmetry events in each list item are complementary in that the first symmetry event implies one oscillator would be faster if its pulse waveform were symmetric while the other symmetry event implies the opposite. Without special considerations, then, these phenomena create problematic alternating indications that each of the oscilla-

tors is faster when the frequencies are close enough that slight asymmetries in the input waveforms are within the measurement capture range of the aforementioned implementations of the invention.

Several additional interesting algebraic relationships are also inherent in the above list of observations. If the frequencies of the two square waves are very close, the sequence will repeat in an alternating pattern for two or more times, and the number of times increases monotonically as the relation between the frequencies approaches identity. Further, each of the conditions in the above list identifies a unique type of relative asymmetry inherent in the pair of waveforms. These facts, and similar ones resulting from other cases (more divergent asymmetry, one waveform essentially symmetric while the other is asymmetric, etc.) may be used to create additional circuitry or algorithms to detect and indicate these events, unique to the asymmetries involved. Additionally, once the asymmetries are detected, the phenomena that would otherwise produce a confused outcome for the circuitry and algorithms can be used to produce definitive frequency comparison outcomes, and even provide additional information about the asymmetries.

An example is provided in Figure 16, which illustrates an adaptation of the symbol-based embodiment of Figures 6a-c designed to handle asymmetric pulse waveforms. Following from the list provided above, a first pair of RS latches are configured to set when a sequence of w_{01} and w_{02} symmetry events, respectively, are observed and reset when another symmetry event is observed. Note that symmetry events w_{01} and w_{02} are complementary in that w_{01} implies oscillator B would be faster if its pulse waveform were symmetric while w_{02} implies oscillator A would be faster if its pulse waveform were symmetric; thus these events would not occur in this sequence if the two input pulse waveforms were symmetric and their frequencies were in a fixed ratio. By taking a logical AND of these two captured events, the first asymmetry signature in the list above would be subsequently indicated. Using a NAND gate to realize this AND operation facilitates use of an additional NAND gate to provide (via DeMorgan's law) an OR operation as well as drive an LED indication of the S_0 asymmetry event. Similarly, three additional pairs of RS latches are responsive to the remaining three pairs of signature symmetry event sequences, each directed to a corresponding NAND gate and LED in the same fashion. All four of the S_0, S_1, S_2, S_3 asymmetry events are thus separately captured and provided with an LED indication, and directed to an OR operation to report the general existence of an asymmetry event.

13 APPLICATIONS

It was mentioned earlier that realizations of the invention can also be used to determine which of two buttons or switches was last cycled on and off. Additional logic circuitry can be added to create a "radio-button" chip that can be ganged to arbitrary numbers of buttons. The technology may also be adapted to a wide range of phase-detection applications. Additionally, the technology appears to have a wide range of other possible applications. These include:

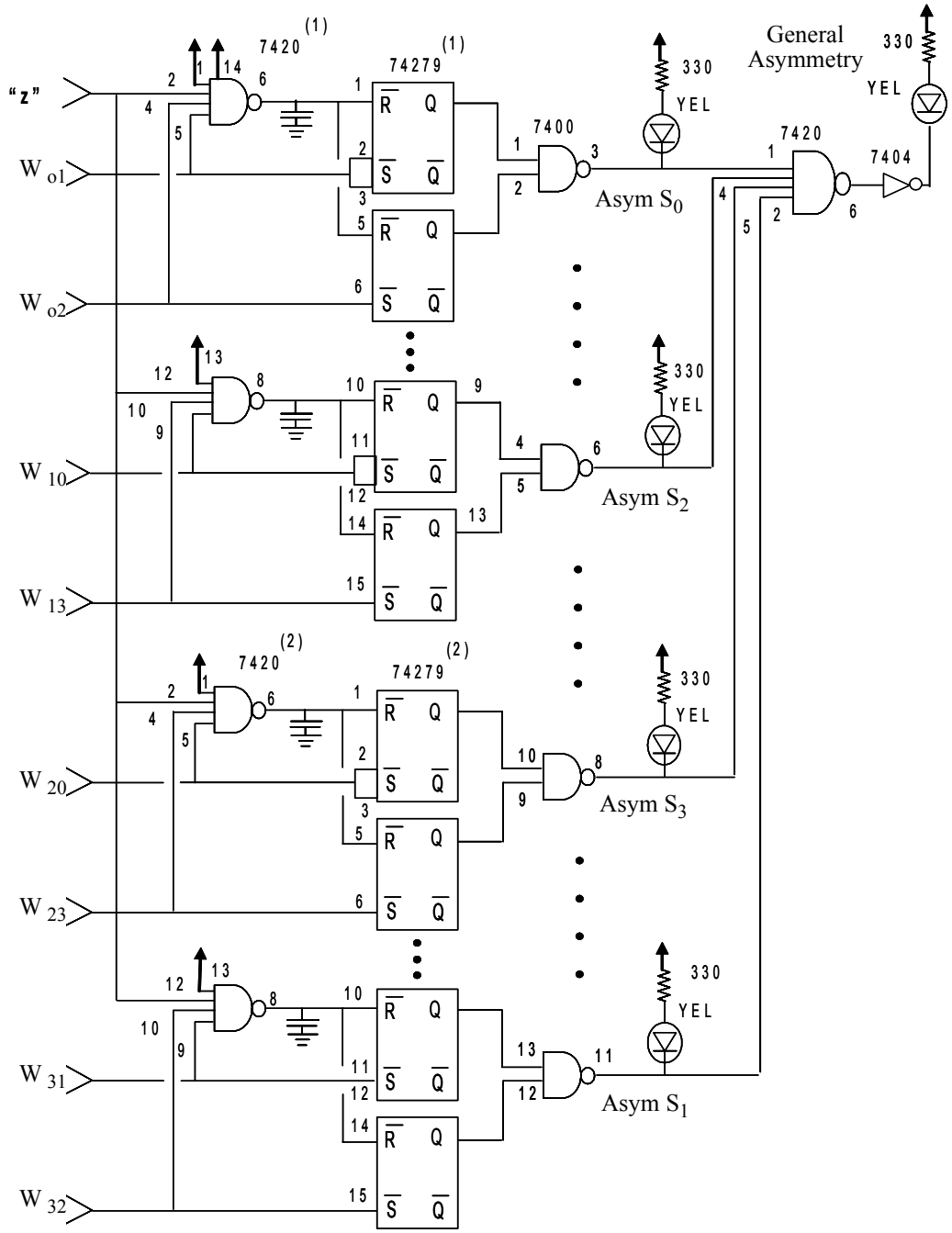


Figure 16

- Synchronous motor control and operation;
- Cross-traffic or resource-constrained timing in transportation systems;
- Resource allocation in manufacturing or project scheduling theory;
- Scheduling of real-time tasks in real-time and near-real-time operating systems;

- Astronomy calculations, perhaps applied in new analyses of ancient archaeoastronomy sites;
- Tool for study of oscillator-coupling phenomena in chaotic and self-organizing systems;
- Geometric lattice design;
- Quantum effects in nanotechnologies;
- Long-duration timing systems as may be used in radioactive waste storage or long-distance space travel;
- Study of molecular vibration;
- Study of energy-transfer among inharmonic periodic modes of oscillation;
- Analysis of biological and ecological systems.

REFERENCES AND ON-LINE LINKS

[1] Lancaster, Don; revised by Berlin, Howard, CMOS Cookbook, Second Edition, Newnes and Howard Sams, Inc., Boston, 1988.

[2] Pending NRI U.S. patent application "Frequency Comparator Utilizing Enveloping-Event Detection Via Symbolic Dynamics Of Fixed Or Modulated Waveforms," details to be provided.

[3] "A Frequency Comparator for Fixed and Modulated Waveforms Utilizing Enveloping-Event Detection: A Theoretical Background Employing a Symbolic Dynamics Framework," NRI Technology Whitepaper, <date??>.

[4] Blaine Quentin Geddes, U.S. Patent 5,939,901, "Synthesizable Flip-Flop Based Phase-Frequency Comparator for Phase-Locked Loops," August 17, 1999.

[5] Walter L. Zweig, U.S. Patent 4,527,080, "Digital Phase And Frequency Comparator Circuit," July 18, 1983.

[6] Ryo Tamaki; Tastuya Kubo, U.S. Patent 6,218,907, "Frequency Comparator And PLL Circuit Using The Same," April 19, 1999.

[7] James B. Rhode, U.S. Patent 4,278,898, "Frequency Comparator For Electronic Clocks," August 13, 1979.

[8] Raymond Huguenin; Hubert Mattey; Jean Engdahl, U.S. Patent 4,608,171, "Frequency Comparator," September 9, 1976.

[9] Jean-Jacques Thiebaut, U.S. Patent 3,947,775, "Phase And Frequency Comparator," April 30, 1974.

.

[10] Henri Butin, U.S. Patent 4,322,686, "Frequency Comparator Circuit," February 27, 1980

[11] Dicy D. Davis, U.S. Patent 3,958,269, "Color Subcarrier Frequency Comparator," August 20, 1974.

[12] Klye L. Burson; Scott O. Campbell; Apparajan Ganesan; Ronald A. Morrison, U.S. Patent 4,677,322, "Frequency Comparator Circuits," August 16, 1984.

[13] Maurizio, U.S. Patent 4,772,852, "Phase-Frequency Comparator for Phase-Locked Loops," September 20, 1988.

[14] Koyo Kegasa, U.S. Patent 4,940,952, "Phase And Frequency Comparator Circuit For Phase Locked Loop," July 10, 1990.

[15] Jean-Claude Abblate; Carl Cederbaum, U.S. Patent 6,563,346 B2,
"Phase Independent Frequency Comparator," May 13, 2003.

[16] Masato Takeyabu; Akira Kikuchi; Toshiyuki Sakai, U.S. Patent 6,392,494 B2,
"Frequency Comparator And Clock Regenerating Device Using The Same," May 21, 2002.

[17] Tsuguhiro Okada; Akira Endo, U.S. Patent 3,987,365, "Digital Frequency Comparator Circuit,"
October 19, 1976.

[18] Yin-Shang Liu; Kuo-Sheng Huang; Hung-Chih Liu, U.S. Patent 6,6501,46 B2,
"Digital Frequency Comparator," November 18, 2003.

[19] Dalius Baranauskas, "Frequency Comparator draws 8 μA," *EDN Access*,
www.e-insite.net/ednmag/archives/1997/030397/05DI_05.htm, March 03,1997.

[20] Edward Foster, "The Phase/ Frequency Comparator Simplified," *Electronics World*,
February 1997 (available at http://ecforster.netfirms.com/phfrqdet/Doc1.html).

[21] W. Dijkstra, "Frequency Window Comparator has Hysteresis, "*EDN Access*,
www.e-insite.net/ednamag/archives/1996/110796/23DI_05.htm, November 7,1996.

[22] (Author unknown), "Frequency Comparator, "*IEC*,
www.geocities.com/IECMaster/circuits_msr/cir_msr011.html, February 15, 2002.