



(22) Date de dépôt/Filing Date: 2000/09/01

(41) Mise à la disp. pub./Open to Public Insp.: 2002/03/01

(45) Date de délivrance/Issue Date: 2005/08/30

(51) Cl.Int.⁷/Int.Cl.⁷ H03K 21/00, G01R 31/3177, H03K 5/14

(72) Inventeurs/Inventors:

ISRAEL, ADRIAN, CA;
DEFAZIO, JODY, CA

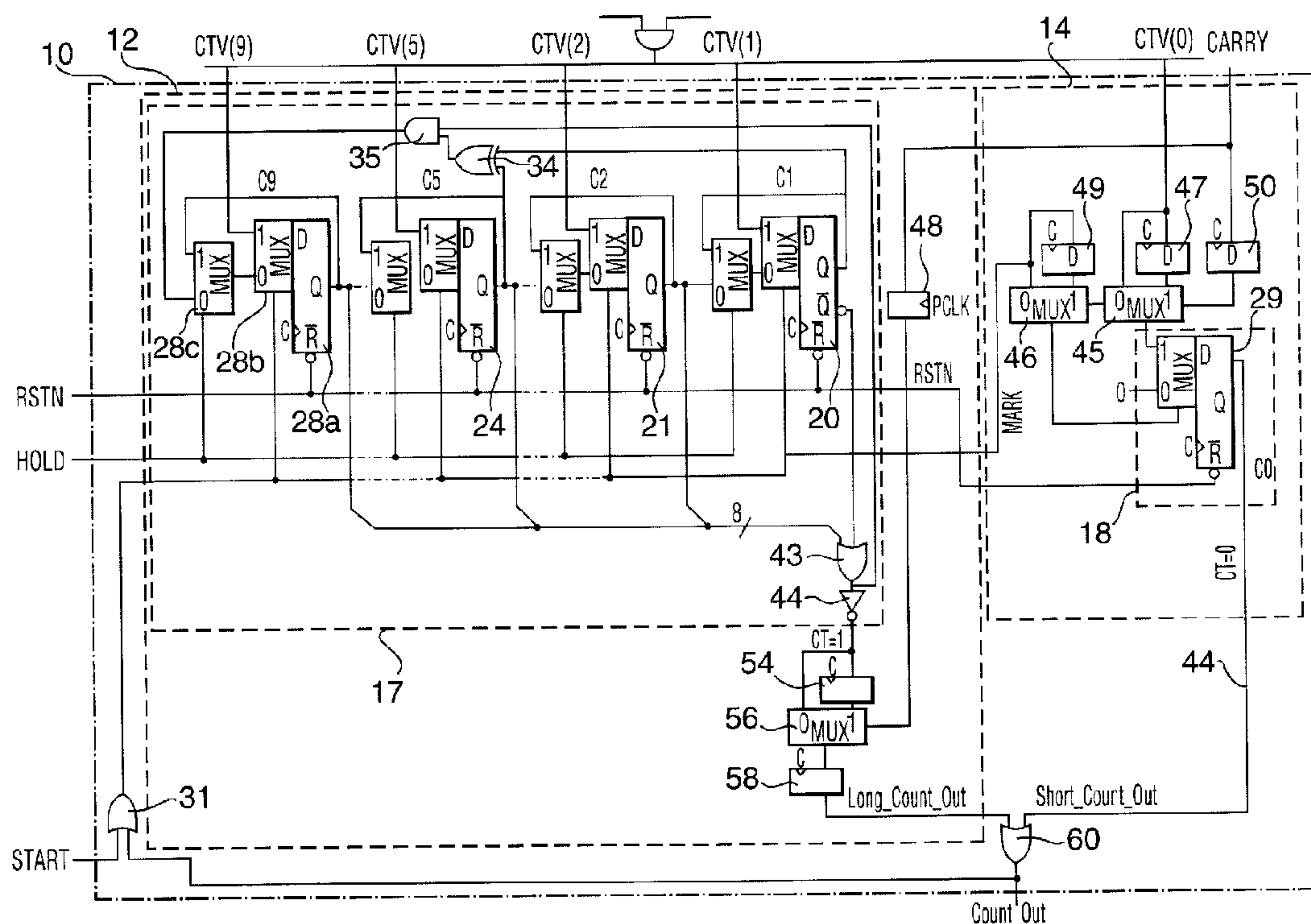
(73) Propriétaire/Owner:

MOSAID TECHNOLOGIES INC., CA

(74) Agent: GOWLING LAFLEUR HENDERSON LLP

(54) Titre : COMPTEUR DE CYCLES HAUTE VITESSE POUR TESTS AUTOMATISES

(54) Title: HIGH-SPEED TEST CYCLE COUNTER



(57) Abrégé/Abstract:

A high-speed wide range counter for Automated Test Equipment applications is divided into two separate, parallel counting blocks for counting over a wide range of count values including short and minimum counts while optimizing gate utilization. The first counting block based on pipeline logic circuitry is used to perform long counts and the second counting block based on wide parallel logic circuitry is used to perform short counts. The two blocks operate one at a time, exclusively of each other. Their outputs are combined to provide a final count value. The overall counter is optimized for high speeds as required in today's ATE applications since the combinatorial logic delay with polynomial counter decoding stages is avoided in the case of short count values. The counter also supports loading on a cycle-by-cycle basis.

ABSTRACT

A high-speed wide range counter for Automated Test Equipment applications is divided into two separate, parallel counting blocks for counting over a wide range of count values including short and minimum counts while optimizing gate utilization. The first counting block based on pipeline logic circuitry is used to perform long counts and the second counting block based on wide parallel logic circuitry is used to perform short counts. The two blocks operate one at a time, exclusively of each other. Their outputs are combined to provide a final count value. The overall counter is optimized for high speeds as required in today's ATE applications since the combinatorial logic delay with polynomial counter decoding stages is avoided in the case of short count values. The counter also supports loading on a cycle-by-cycle basis.

HIGH-SPEED TEST CYCLE COUNTER

FIELD OF THE INVENTION

The invention relates to test cycle generation and more specifically to a high-speed polynomial counter used to generate test cycle timing values in automated test equipment (ATE).

BACKGROUND OF THE INVENTION

Automated test equipment used for testing memories, ASICs, microprocessors and other types of digital circuits requires the generation of test cycles having variable durations in order to examine the performance of the device under test (DUT) in different operating modes during predetermined timing intervals. A test cycle essentially consists of a portion of time during which a certain test operation is performed on the DUT. One ATE feature illustrating this requirement is to provide the user the ability to change the duration of a test cycle on a cycle-by-cycle basis. For example, data can be written to the device under test at one frequency and read out at a different frequency, requiring the ATE to be able to generate different durations for each test cycle.

Generating such variable duration test cycles requires varying degrees of timing resolution. Essentially, a test cycle consists of two portions, a coarse timing portion requiring coarse timing resolution and a fine or precise portion requiring fine timing resolution. The coarse portion of the test cycle is typically implemented using a counter which counts an integer number of system clock cycles. The fine portion of the test cycle is typically implemented using additional circuitry that accumulates fine delay

portions and generates a carry signal when the accumulated result of fractional clock fine delays is greater than one system clock period.

With reference to the coarse portion of the test cycle timing, since ATEs typically operate over a wide range of frequencies, wide range counters are needed to support timing generation in ATE applications. The counters must not only have a wide range to support a multitude of testing frequencies but they must also be quickly re-loadable in order to allow the user to redefine the test cycle period on a cycle-by-cycle basis with no "dead" time. Typically, synchronous counters operating in these applications can be reloaded at the maximum rate of one reload every 2 system clocks where one clock cycle is needed to load the counter and one clock cycle is needed to perform the actual count. In the case of a minimum count, i.e. a count of 0, the counter would require 2 system clock cycles to perform the minimum count.

It has been found that using polynomial counters to provide an integer number of clocks representative of the coarse delay portion of a test cycle is optimal for ATE applications due to the structure of such counters. Polynomial counters, otherwise known as linear feedback shift registers (LFSRs), can operate at much higher frequencies than binary counters. Typically, binary counters require multiple stages of combinatorial logic circuits to resolve carry propagation, whereas polynomial counters use a shift register implementation and a feedback path which together can reliably count at much higher frequencies. In a conventional polynomial counter, a feedback path is taken from two or more predetermined tap positions (depending on the number of register stages) and combined in an exclusive OR function to be feedback to the input of the first stage of the shift register to produce a pseudo-random polynomial count

sequence. An n-bit polynomial counter is reset to an initial valid count value (typically 1000...00) and sequences through a series of (2^n-1) pseudo-random states (typically ending at 0000...01) and circulates continuously through all valid states. A 'No_Op' (no operation) state is specifically avoided although such a state (typically defined by all 0's) can be loaded into the counter to generate a 'No_Op'.

Polynomial counters are typically used as pre-scalar counters, the outputs of which are usually applied to slower binary counters. The outputs of the shift register portion of a polynomial counter are typically decoded by combinatorial logic. This decoding logic introduces significant delay which at high speeds approaching 1GHz, as required in today's ATE applications, begins to limit the operating speed of the counter. For example, as long as the operating frequency of the ATE is relatively low (less than 100MHz), using a polynomial counter having an n-bit timing input will generate a count having (2^n-1) unique states within the required test cycle window. However, as the frequency of the ATE is increased significantly (over 100MHz to up to 1GHz), as is the case in today's high speed applications, the output of a polynomial counter can no longer keep up with the speed of the test cycle pipeline due to the polynomial counter decoding logic – especially in the case of low count values. For example, for a count of 0 or 1, the combinatorial logic associated with the output of the polynomial counter would take too long to decode the count result in order to provide a "true 0 count" to the system within one test cycle pipeline stage. As a result, one or more additional pipeline stage(s) must be inserted on the count output of the polynomial counter. For example, in case where one pipeline stage is added, this additional pipeline stage adds one clock count to the polynomial counter sequence which previously would count over the range

(2^n-1):0 and now will count over the range 2^n :1. Therefore, the polynomial counter is no longer able to count the required minimum count value representing the required output clock latency.

Although conventional polynomial counters are attractive for their high speed
5 operating characteristics, they are nonetheless somewhat rigid in applicability, having one less count than a binary counter with the same number of bits, and counting through a (pseudo-random) polynomial sequence rather than a more convenient binary sequence. Furthermore, the decoding logic associated with generating a counter output begins to limit the operating speed of polynomial counters at high testing speeds,
10 especially for low count values.

Accordingly, there is a need for a high speed wide range test cycle counter which can be loaded on a cycle-by-cycle basis and which ensures maximum gate utilization.

15 SUMMARY OF THE INVENTION

The present invention seeks to overcome the disadvantages of the prior art associated with high-speed test cycle counters.

The test cycle counter of the present invention is comprised of a long cycle counter and a short cycle counter both of which receive test cycle timing value inputs as
20 well as a carry input from fine delay circuits. The long cycle counter performs long count sequences and the short cycle counter performs short count sequences including a minimum count.

In accordance with an aspect of the present invention, there is provided a counter responsive to timing input signals and a carry signals for providing a count output signal, which includes: (a) a first counting block responsive to a first group of timing signals providing a first count sequence for generating a first count signal upon completion of stepping through the first count sequence; (b) a second counting block responsive to a second group of timing signals providing a second count sequence for generating a second count signal upon completion of a stepping through the second count sequence; (c) a combining block for combining the first and second count signals and generating a feedback signal for enabling loading of the first and second counting blocks with new values for the first and second groups of timing signals; (d) a first selecting block for selectively delaying the first count signal by a predetermined number of clock delays in response to the carry signal; and (e) a second selecting block for selectively delaying the second count signal by the predetermined number of clock delays in response to the carry signal.

In accordance with a further aspect of the present invention, there is provided a counter for receiving a carry signal indicative of carry information and providing a counter output equivalent to an integer number of system clock periods. The counter output is generated in response to either one of a long continuing sequence or a short counting sequence. The short counting sequence includes a minimum count. The counter includes: (a) a long count block for stepping through the long counting sequence in response to a first group of timing value inputs and for generating a long count output signal; (b) a short count block for stepping through the short counting sequence in response to a second group of timing value inputs and for generating a short count output signal; (c) an output block for receiving either one of the long output signal or the short output signal and for generating a feedback signal for alternately loading new timing value inputs into either one of the long count block or the short count block; and (d) a first delay means and a second delay means for selectively delaying either one of the long count signal or the second group of timing value inputs by a predetermined number of system clock periods in response to the carry signal.

In accordance with a further aspect of the present invention, there is provided a counter responsive to timing input signals and a carry signals for providing a count output signal, which includes: (a) a first counting block responsive to a first group of timing signals providing a first count sequence for generating a first count signal upon completion of stepping through the first count sequence; (b) a second counting block responsive to a second group of timing signals providing a second count sequence for generating a second count signal upon completion of a stepping through the second count sequence; (c) a first delay block receiving the first count signal for generating a first delayed count signal; (d) a second delay block receiving the second count signal for generating a second delayed count signal; (e) a first selecting block receiving the first count signal and the first delayed count signal for selectively providing either the first count signal or the first delayed count signal as a first output count signal to a combining block; (f) a second selecting block receiving the second count signal and the second delayed count signal for selectively providing either the second count signal or the second delayed count signal as a second output count signal to the combining block; and (g) the combining block for combining the first and second output count signals and generating a feedback signal for enabling loading of the first and second counting blocks with new values for the first and second groups of timing signals.

BRIEF DESCRIPTION OF THE DRAWINGS:

Fig. 1 is a block diagram of a high-speed wide range counter according to an embodiment of the invention;

Fig. 2 is detailed schematic illustrating a preferred embodiment of the invention;

Fig. 3 is a timing diagram of the embodiment of Fig. 2;

Fig. 4 is a table illustrating all the count states of the counter according to an embodiment of the invention.

DETAILED DESCRIPTION OF THE INVENTION

There are two factors which limit the speed of operation of a polynomial counter: the combinatorial logic used to decode the outputs of the shift register stages into a single output, and the feedback path used to drive the counter through its sequence of valid states. Of these two, the combinatorial logic generates the longest delay. When the speed of operation of a counter is very high, as in the case of counters used in automatic test equipment (ATE), the delay required to properly operate the polynomial counter and produce the correct output result becomes too large compared to the window of operation allocated for the counter to complete this task by the test cycle pipeline control. As a result, there is a need to insert an additional pipeline delay stage after the output of the counter. This additional pipeline stage delays the output of the polynomial counter by one additional count. If a short count value is required for a particular test cycle, for example a count of 0 or 1, as previously mentioned, the combinatorial decoding logic of a conventional polynomial counter becomes the critical path of the counter, limiting its speed of operation. According to an embodiment of the invention, one way of recapturing such short count cycles is to separate the overall counter into two parallel counting blocks operating separately where one long cycle counting block is responsible for long counts where the combinatorial logic associated with generating its output is not a constraining issue, and a short cycle count counting block which counts short counts and operates much more quickly than the long cycle counting block by by-passing the large combinatorial logic of the long count counting block.

Fig. 1 is a simplified block diagram of an n-bit high-speed wide range polynomial counter **10** according to an embodiment of the invention. The counter **10** includes a first counting block **12** (designated as Long Cycle Counter or LCC) and a second counting block **14** (designated as Short Cycle Counter or SCC). The first counting block **12** comprises an (n-m)-bit counter **17**, and a carry circuit **9** comprising a latch **13a**, and a multiplexer **15a**. Counter **17** receives as inputs test cycle timing value signals CTV[(n-1) : m], i.e. the most significant bits, a reset signal RSTN, a synchronization signal MARK, a test cycle freeze signal HOLD, and the system clock signal CLK. The RSTN signal is an asynchronous signal which is used to reset the counter stages. The MARK signal defines the timing boundaries of a test cycle by marking the loading (on the next system clock rising edge) of a new test cycle timing value into counter **17**. The HOLD signal allows an ATE user-initiated asynchronous freeze of counter **17** when asserted. Such a HOLD function is useful in testing data cell retention for example, whereby test data would typically be written into selected cells and in response to the HOLD signal being asserted, the test condition would be frozen for a predetermined time to analyze data retention. The latch **13a** receives as an input the output of counter **17** and is clocked by the system clock CLK. The output of the counter **17** can be delayed by one clock based on the state of a CARRY signal generated by a fine delay portion (not shown). The latch **13a** in conjunction with the multiplexer **15a** are used to implement the CARRY function within the pipeline stages where the counter **17** can be considered to be part of pipeline stage p and the carry circuit **9** can be thought of as being part of pipeline stage p+1. Generally, when the CARRY signal is logic '1', the latch **13a** introduces one clock delay and provides the output of counter **17** to the multiplexer **15a** one clock delay after

its occurrence. The other input of the multiplexer **15a** is taken from counter **17** directly without an additional clock delay, thereby allowing the multiplexer **15a** to selectively add one clock delay to the output of counter **17** based on the CARRY input. The CARRY input is provided to the multiplexer **15a** through a pipeline latch **19**, which is clocked by
 5 a pipeline clock PCLK. The pipeline clock PCLK, as will be described in further detail below, is a pulse which is derived from the MARK signal. The latch **19** is used to align the pipeline CARRY to the output of the counter, i.e. the previous pipeline stage CARRY is aligned with the current pipeline stage. If CARRY is logic '1', then the output of counter **17** is delayed by one pipeline stage, if CARRY is logic '0', then the output of
 10 counter **17** is not delayed and instead provided directly to the combining circuit **16**. For speed performance considerations, the long combinatorial logic delay path of the Long Cycle Counter is realigned to the system clock CLK using the latch **13e**.

A second counting block **14** includes an m-bit counter **18** which can be implemented using another polynomial counter, and a second carry circuit **8** comprising
 15 two multiplexers **15b** and **15c** and three latches **13b**, **13c** and **13d**. The purpose of the second counting block **14** is to rapidly provide the remainder of the full count which cannot be provided by the first counting block **12**. Whereas the first block **12** receives test cycle timing values $CTV[(n-1) : m]$, the second block **14** receives cycle timing values $CTV[(m-1) : 0]$, i.e. the least significant bits. Preferably, m has a value which is
 20 significantly smaller than n and m should be as small as possible, for example, $m \leq 4$. As can be seen, the carry circuit **8** in this case is implemented in a pipeline stage prior to the m-bit counter, i.e. before the counter is loaded. As described with reference to the first counting block **17**, the second counting block **14** can also be thought of as being

distributed over two pipeline stages: the carry circuit **8** is implemented in pipeline stage p and the m -bit counter is implemented in pipeline stage $p+1$. The m -bit counter **18** receives as inputs the m -bit count value either delayed by one clock through latch **13d** or provided directly from test cycle timing value inputs $CTV[(m-1) : 0]$ depending on the

5 state of the CARRY input. The CARRY input passes through a delay stage **13b** which is clocked by the system clock CLK. If CARRY is logic '1', the cycle timing value inputs are delayed one clock delay through **13d**. Alternately if CARRY is logic '0', then the cycle timing value inputs are taken directly as inputs into the multiplexer **15c**. Counter **18** also receives the RSTN reset signal as an input for asynchronously resetting the

10 count value. In addition, the counter **18** receives the MARK synchronization signal either delayed by one clock through latch **13c** or passed directly through multiplexer **15b**. Optionally, a HOLD signal can also be input into the second counting block **14** (although according to a preferred embodiment, the HOLD signal is not provided to the second counting block due to speed considerations; i.e. the second counting block has

15 a small number of bits resulting in a small count value, thereby making a hold function not practical since HOLD and MARK cannot be asserted simultaneously). The m -bit counter **18** is clocked by the system clock CLK and provides its Short_Count_Out output to the combining circuit **16**. The multiplexers **15b** and **15c** are controlled by the output CARRY signal similarly to the way multiplexer **15a** was also controlled by the

20 CARRY signal, i.e. through a pipeline latch. The CARRY signal must be stable for the entire counter loading period so a delay of one clock through latch **13b** is inserted to ensure this occurs. Therefore multiplexers **15b** and **15c** are controlled by the CARRY signal status latched in latch **13b**. If the CARRY signal is logic '1' indicating that a carry

has occurred, then multiplexers **15b** and **15c** select the MARK and CTV inputs which are delayed by one clock delay, **13c** and **13d** respectively. On the other hand, if CARRY is logic '0', then multiplexers **15b** and **15c** select the MARK and CTV inputs which are not delayed by a clock delay. It should be noted that the CARRY signal is provided as an input to both the LCC **12** and the SCC **14**.

As can be seen, the m-bit counter **18** has the carry function implemented before loading in order to reduce the combinatorial logic delay accumulated by the Short_Count_Out signal. Consequently, the loading of the MARK signal and CTV[(m-1 : 0)] are conditioned by the CARRY signal. If CARRY is logic '1' these signals are delayed by one clock, if CARRY is logic '0' these signals are not delayed.

A gate connected to the on the input of the counter **10** can be used to asynchronously stop the counter **10**. For example, the CTV inputs can be combined in an AND gate **37** with an active logic '0' STOP signal as shown in Fig. 1.

It should be noted that the first and second counting blocks **12** and **14** respectively cannot operate simultaneously since they are intended for two different count functions. The first block **12** is intended for long count values while the second block **14** is intended for short or minimum count values. Therefore the second counting block **14** must have a counting range greater than or equal to the minimum counting range of the first counting block **12**; for example according to a preferred embodiment where $n=9$, $m=1$ the full counter range is [511:0] with the first counting block counting over the range [511:1] and the second counting block counting over the range of [0], i.e. a 1-bit counter. In terms of actual system clock delays, since a minimum count=0 takes 2 system clocks to execute, the range of actual system clock cycles for the counter

would be [513:2] with the first counting block counting [513:3] clocks and the second counting block counting 2 clocks.

The general operation of the high speed test cycle counter **10** of **Fig. 1** will now be described. The RSTN reset signal clears the contents of the two counting blocks **12** and **14**, allowing the ATE user to load/program a new test cycle. New cycle timing value inputs are loaded into the counting blocks **12** and **14**. The START signal is asserted for one full system clock cycle and counting begins. For counting block **12** which contains the $[(n-1):m]$ -bit polynomial counter **17**, the counting proceeds as previously described, by stepping through a pseudo-random sequence of valid states through the clocked shift register stages based on the cycle timing value inputs CTV. If the user wishes to freeze the test sequence at any time, the HOLD signal is asynchronously asserted and the count sequence is stopped for the specified number of HOLD cycles. By deactivating HOLD, the count sequence is resumed.

As the counter **17** continues through its polynomial count sequence, it will eventually get to the second last state in the sequence. As will be described in more detail later, this second last state is used to force the polynomial counter **17** into a state which effectively stops the count from resuming again, unlike most conventional polynomial counters which will continue to proceed through the count sequence repeatedly until stopped by external intervention. Once counter **17** has reached the final state in the sequence, it outputs a signal to the pipeline latch **13a** which is clocked through to the multiplexer **15a** on the next rising edge of the system clock CLK. The last state is also provided directly as the other input to multiplexer **15a**. Based on the state of the CARRY signal and the pipeline clock PCLK, one of the two inputs of

multiplexer **15a** is provided to the combining circuit **16** on the next rising edge of the system clock CLK. This sequence of operational steps is taken if the period timing value inputs specify a count between (n-1) to m, i.e. a long cycle count.

Alternately, for a short cycle count, i.e. a count in the range (m-1) : 0, counting
 5 block **14** is used. As in the case of the first counting block **12**, the RSTN reset signal clears the contents of the second counting **14**, allowing the ATE user to load/program a new test cycle. New cycle timing value inputs CTV[(m-1) : 0], i.e. the least significant bits are loaded into the counting blocks **12** and **14**. The START signal is asserted and counting begins. Both the MARK signal and the CTV inputs are conditioned on the
 10 state of the CARRY input. If CARRY is logic '0', the MARK and CTV inputs are passed directly to multiplexers **15** and **15c** which subsequently provide the m-bit counter **18** with its short count value. Alternately, if CARRY is logic '1', both MARK and CTV inputs are delayed by one clock delay and the m-bit counter **18** receives its count value one clock delay later. The system clock CLK synchronizes the input of the m-bit value from
 15 multiplexer **15c** on the next rising edge of the system clock CLK. The output of counter **18**, is then provided to the combining circuit **16** on the next clock rising edge. The output of the combining circuit **16** is combined by NOR gate **31** with the START signal to produce the MARK signal which is feedback to both the first and second counting blocks **12** and **14** to indicate the beginning of the next count sequence. In the absence
 20 of the MARK signal, i.e. for the first count sequence, the START signal alone initiates the counting sequence.

It should be noted that the separation of the overall counter into two separate, parallel counting blocks allows the handling of a wide range of count values including

short, minimum counts while optimizing gate utilization. By using the first counting block based on pipeline logic gates to implement long counts and the second counting block based on wide parallel logic gates to implement short counts and then combining the output to provide a final count value, the overall counting function is optimized for high speeds as required in today's ATE applications since the combinatorial logic delay with polynomial counter decoding stage is avoided in the case of short count values. This implementation also allows the counting blocks to be loaded on a cycle-by-cycle basis since the feedback path provided by the MARK signal provides the counting blocks with new timing values only after the current count is completed.

A more specific structure of the first and second counting blocks **12** and **18** will now be discussed in more detail with reference to **Fig. 2**. According to a preferred embodiment of the invention, the high-speed counter **10** comprises the first counting block **12** and the second counting block **14**. The first counting block **12** is implemented using a polynomial counter **17** comprising $[(n-1):m]$ shift register stages. In a preferred embodiment, with $n=9$ and $m=1$, counter **17** has 9 stages, starting with stage 9 **28**, and ending with stage 1 **20** and therefore is capable of counting over the range $[(2^n - 1):m]$ or $[511:1]$. In terms of actual clock cycle delays, given the 2 clock delay for the minimum count, the actual clock cycle range becomes $[513:3]$ clocks. Each of these polynomial counter stages (only 4 of the 9 stages are shown) **28**, **24**, **21**, **20** comprise an input multiplexer controlled by the MARK signal and a conventional D-type flip-flop clocked by the system clock CLK and reset by the RSTN signal. Taking one of the stages as an example, stage **28** consists of D-type flip-flop **28a**, an input multiplexer **28b** and a second multiplexer **28c**. The two multiplexers essentially operate as a 3:1

multiplexer with the following possible conditions: normal LFSR operation (i.e. count sequence), load new cycle timing value, hold count. More specifically, the input multiplexer **28b** passes either the test cycle timing value input CTV[9] to load a new cycle value or provides a feedback Q output signal C9 of the flip-flop **28a** through
 5 multiplexer **28b** under the control of the MARK signal; i.e. when MARK is logic '1', the new cycle timing value input CTV9 is provided to the D-input, and if MARK is logic '0', the output of the second multiplexer **28c** is provided to the D-input as part of the normal LFSR count sequence. The second multiplexer **28c** provides as its output either the feedback version of the Q-output of the flip-flop **28a** or, in the case of the last stage **28** of
 10 the counter, the multiplexer **28c** takes the output of a feedback block comprising an AND gate **35** and an exclusive OR gate **34** according to conventional polynomial counter feedback connection practices. Multiplexer **28c** is controlled by the HOLD signal which allows values stored in the shift register stages to be frozen for a duration selected by the user. The output of stage **28**, i.e. the Q-output of the D-type flip-flop **28a**
 15 is provided to a multi-input combining gate which in this implementation is shown as an OR gate **43** as is also provided as an input to the next counting stage **24**. All stages in counter **17** have the same structure as stage **28** with the exception of the first stage **20** in which case the Q\ output of the D-type flip-flop **20a** is also used in addition to the Q-output. In this case, it is the Q\ output which is provided to the combining gate **43** rather
 20 than the Q-output as was the case with all the other counter stages. By providing the Q\ output to the combining gate **43**, the counter **17** is able to detect the second last count state prior to count completion. In other words, instead of allowing the counter **17** to continue directly to the final count value and then begin the count cycle again as a

conventional polynomial counter would do, counter **17** uses the second last state to signify the upcoming end of count and feeds this information back to the last stage of the counter to prevent the counter from beginning to cycle through the count sequence once again. More specifically, AND gate **35** inserted in the feedback path is used to

5 allow counter **17** to detect the second last state in the count sequence and stop counting (after which, the short cycle counter will take over to count the last sequence). For example in this case, the second last count state will occur when all outputs of the shift register stages except the last one, i.e. stage **20** are all logic '0'. Since the Q\ output is taken from stage **20** instead of the Q output, this output will also be logic '0'.

10 Therefore, all inputs into the OR gate **43** are logic '0'. This condition is feedback into AND gate **35** to produce a logic '0' output which is then fed into the '0' input of multiplexer **28c**. Once this state has been detected, counter **17** forces the last sequence, which in this case is designated by code '0x0' (in hexadecimal format – a more detailed description of the count sequence codes will be described later). This

15 No_Op state is usually avoided in conventional polynomial counter design. However, according to the invention, this No_Op state is intentionally loaded into the long cycle counter while passing the remaining counting responsibility to the short cycle counter. Therefore, one novel feature of the invention is that the No_Op state is loaded to stop the counting sequence of the LCC and allow the remainder of the count to be handled

20 by the SCC. As a result, the two counting blocks can operate one at a time to optimize the operation of the entire counting system.

Continuing with the description of **Fig. 2**, the output of the combining OR gate **43** is inverted through inverter **44** to provide an active logic '1' signal and provided as the

output of counter **17** to a latch **54** which introduces one clock delay as described with reference to Fig. 1. The output of combining gate **43** is also provided to the AND gate **35** of the feedback block. The other input of the AND gate **35** is provided by the exclusive OR gate **34**, which in turn receives its two inputs from the fifth and first stages
5 of the counter **17** which are selected to provide the appropriate count operation for the 9 bit counter implementation. The latch **54** is clocked by the system clock CLK and provides its output to multiplexer **56** which under control of pipeline latch **48** containing the pipeline CARRY information, selects between the output of inverter **44** i.e. the direct Count_Out or latch **54** the Count_Out delayed by one system clock CLK. The output of
10 multiplexer **56** is provided to a second latch **58** which generates the Long_Count_Out signal which in turn is provided as one of the inputs to the OR gate **60**. The other input of OR gate **60** is the Short_Count_Out signal which is provided from the second counting block **14** as described in Fig. 1. More specifically, with reference to Fig. 2, with $m=1$, the 1-bit counter **18** is implemented by a D-type flip/flop plus multiplexer combined
15 circuit **29** which has the same structure as all the other count stages in the first counting block **17**. Multiplexer **46** and **45** correspond to multiplexers **15b** and **15c** respectively in Fig. 1 and pipeline latches **49**, **47**, and **50** correspond to pipeline latches **13b**, **13c** and **13d** of Fig. 1 respectively.

The operation of the preferred embodiment of the invention illustrated in Fig. 2
20 will now be described with reference to timing diagram shown in Fig. 3 and with reference to pseudo-random codes table shown in Fig. 4. A number of representative test cycles are shown (cycle 1-cycle 7) all having different durations and different requirements. For example, while cycles 1-4 are comprised of different user-specified

system clock delays, cycles 5-7 also comprise user-specified hold and carry instructions. Fig. 3 illustrates the various transitions and states of the control signals and counter inputs and outputs for these test cycles. The system clock CLK is a continuous free-running clock which runs at a predetermined frequency. The START, and HOLD signals are provided by ATE user input. The CARRY signal is provided as a feedback from fine delay circuits (not shown) indicating the accumulation of fine delays equal to one full system clock delay.

Prior to the beginning of cycle 1, the START signal is asserted by ATE user input for one full system clock and the cycle timing values for cycle 1, i.e. the minimum count value count=0 (requiring a minimum of 2 system clocks, i.e. 2CLK and designated as 1 in hexadecimal format as shown in Fig. 4) is provided to the input of the counter. As previously explained, this minimum count is handled by the second counting block 14. The MARK signal, generated by the START signal through OR gate 31, marks the beginning of cycle 1 and, on the first subsequent rising edge of the system clock CLK following the beginning of cycle 1, the counter is loaded with the test cycle timing value of 1 and begins stepping through the count sequence, as shown by the Counter Status timing line. The "1" status is the first step in the sequence followed by "0", the final step in this sequence. The pipeline clock pulse PCLK is generated one clock delay after assertion of the MARK signal. The PCLK clocks the CARRY information into the current counting cycle. In the case of cycle 1, the CARRY signal is logic '0' so it has no effect on the count. On the second rising edge of the system clock CLK after the beginning of cycle 1, the counter provides the completed count signal Count_Out to subsequent circuitry in the pipeline as well as to NOR gate 31. The feedback path

through NOR gate 31 reasserts the MARK signal marking the beginning of the next test cycle, cycle 2.

Cycle 2 has a count value count=1 (requiring 3CLK delays as shown in Fig. 4, which is designated as 2 in hexadecimal format). This desired count value is loaded into the counting block 14 on the rising edge of the system clock following the assertion of the MARK signal. Stage C9=0, ... stage C5=0, stage C2=0 and stage C1=1, all generate a logic "1" on the output of OR gate 43. One system clock later, Long_Count_Out goes to a logic "1", making Count_Out go to a logic "1" through OR gate 60 and eventually MARK go to a logic "1" through OR gate 31. Pipeline clock pulse PCLK, is once again generated on the next rising edge of the system clock CLK following the assertion of the MARK signal, thereby providing the CARRY for the current pipeline stage. The counter once again steps through its sequence of count states, 2,0,0 in this case as shown in the Counter Status timing line. Once the last count is detected, AND gate 35 is turned off thereby turning off the Long Cycle Counter. Once the final count state has been counted by the Short Cycle Counter, the Count_Out signal is asserted which in turn reasserts the MARK signal and begins cycle 3.

Cycle 3 is identical to cycle 1, requiring a count=0 to be performed as previously described except that the cycle is initiated by the rising edge of MARK rather than the START signal. The first clock in cycle 3, as in all cycles, is required to generate the rising edge of MARK.

In cycle 4, a count value count=2 is requested by the test cycle timing inputs (corresponding to 4 in hexadecimal format and taking 4CLK to complete as indicated in Fig. 4). As with the preceding cycles, the rising edge of the MARK signal initiates cycle

4 and on the rising edge of the system clock following the assertion of the MARK signal, the counter loads the new CTV value. Pipeline clock pulse PCLK, is once again generated on the next rising edge of the system clock CLK following the assertion of the MARK signal, thereby providing the CARRY for the current pipeline stage. The counter
 5 steps through the counting sequence, 4,2,0,0 in this case, as shown in the Counter Status timing line. Once the counter has completed its count, which in this case is performed by the first counting block 12, the MARK signal is reasserted and cycle 5 begins.

Cycle 5 comprises a count value count=2 (4 in hexadecimal) and 2 HOLD
 10 instructions. The MARK signal initiates cycle 5 and the Counter Status line provides the corresponding counter states which in this case start off with 4 followed by 2. At that point, the 2 HOLD instruction causes the counter to hold the current state for 2 clock cycles, creating two more states of 2 on the Counter Status line. After the 2HOLD has been completed, HOLD is de-asserted and the counter continues to finish the 4 CLK
 15 count, by stepping to the final two count states 0, 0 as shown in the Counter Status timing line. The Count_Out output is pulsed at the end of the cycle 5 and reasserts the MARK signal, thereby initiating cycle 6. It should also be noted that a CARRY signal is asserted during cycle 5 in preparation for the next cycle, cycle 6 which requires a CARRY.

20 Cycle 6 begins much the same as has been already discussed with a count value count=0, requiring 2CLK cycles. However, since a CARRY has been input as part of this test cycle, loading of the count value is delayed by one clock cycle, shown in the Counter Status line by an additional 1 state. This is because since the count=0 is

performed by the short cycle counter block 14, the CARRY signal is provided to the input stage of that block, i.e. latch 50 and delays the input of the CARRY signal by one clock delay. On the third rising edge of the system clock CLK after the beginning of cycle 6, the Counter Status timing line shows the value 1, i.e. the first step in the count=0 sequence, which is followed by 0 which is the second and final state in the count=0 sequence. Once the count completes, the Count_Out signal is asserted which reasserts the MARK signal signifying the beginning of cycle 7.

Test cycle 7 consists of a count=2 and a CARRY. The counter proceeds much the same as described above with reference to cycle 6 except that in this case, the CARRY signal is applied after the counter (first counting block 12) has completed its count to multiplexer 56. As shown in the Counter Status timing line, the count sequence 4,2,0,0 corresponding to count=2 (4 in hexadecimal format requiring 4CLK to complete) is followed by the CARRY delay, shown as an additional 0 state. Once the count is complete the Count_Out signal is asserted which reasserts MARK thereby beginning the next cycle (not shown). Note that for CT=2, the sequence proceeds from state 4 to 2 and when state 2 is detected, the remaining state, i.e. 1 is then subsequently counted by the short cycle counter. In general, all the states in the counter sequence beyond 2 are handled by the long cycle counter and the last state, 1 is handled by the short cycle counter, as shown in Fig. 4. The two counting blocks LCC and SCC therefore operate in an alternating mode.

It is also important to note that in the last two cycles, cycle 6 and cycle 7, the CARRY was handled differently as described with reference to Fig. 2. In the case of cycle 6, since the count required was a minimum count, the CARRY was applied to the

input of the counter, thereby by-passing combinatorial logic delays associated with the first counting block 12. On the other hand, in the case of cycle 7, since the count required was larger than the minimum count, the CARRY was applied after the output of the first counting block 12. In this manner, both long and short counts can be flexibly
5 and rapidly processed using maximum gate utilization.

Although only 7 test cycles have been illustrated, it can be appreciated that similar cycles and corresponding signal behaviour would arise given the many other count states outlined in Fig. 4.

A high speed wide range counter for ATE applications comprising two counting
10 blocks has been described. The counter includes a first counting block responsible for handling long count values and a second counting block responsible for handling short count values including a minimum count. The counter receives a carry input from fine delay circuitry. The first counting block is based on pipelined logic and combinatorial circuitry whereas the second counting block is based on wide parallel logic circuitry.
15 Only one of the two counting blocks are active at any time and their outputs are combined to provide an overall counter output.

The embodiments of the invention described above are provided only as illustrations and should not be construed as limiting the scope of the present invention. Numerous modifications and improvements may occur to those skilled in the art.
20 Therefore, it is to be understood that this invention is not limited to the illustrated embodiments but rather any number of alternate embodiments which are covered by the appended claims.

What we claim is:

1. A counter responsive to timing input signals and a carry signals for providing a count output signal comprising:
 - (a) a first counting block responsive to a first group of timing signals providing a first count sequence for generating a first count signal upon completion of stepping through said first count sequence;
 - (b) a second counting block responsive to a second group of timing signals providing a second count sequence for generating a second count signal upon completion of a stepping through said second count sequence;
 - (c) a combining block for combining the first and second count signals and generating a feedback signal for enabling loading of the first and second counting blocks with new values for said first and second groups of timing signals;
 - (d) a first selecting block for selectively delaying said first count signal by a predetermined number of clock delays in response to said carry signal; and
 - (e) a second selecting block for selectively delaying the second count signal by the predetermined number of clock delays in response to said carry signal.
2. The counter of claim 1, wherein said first counting block is comprised of an n-stage linear feedback shift register.
3. A counter for receiving a carry signal indicative of carry information and providing a counter output equivalent to an integer number of system clock periods, said counter output being generated in response to either one of a long continuing sequence or a short counting sequence, said short counting sequence including a minimum count, said counter comprising:

- (a) a long count block for stepping through said long counting sequence in response to a first group of timing value inputs and for generating a long count output signal;
- (b) a short count block for stepping through said short counting sequence in response to a second group of timing value inputs and for generating a short count output signal;
- (c) an output block for receiving either one of said long output signal or said short output signal and for generating a feedback signal for alternately loading new timing value inputs into either one of said long count block or said short count block; and
- (d) a first delay means and a second delay means for selectively delaying either one of said long count signal or said second group of timing value inputs by a predetermined number of system clock periods in response to said carry signal.

4. The counter of claim 3, wherein said long count block is comprised of a linear feedback shift register having n stages.

5. A counter responsive to timing input signals and a carry signals for providing a count output signal comprising:

- (a) a first counting block responsive to a first group of timing signals providing a first count sequence for generating a first count signal upon completion of stepping through said first count sequence;
- (b) a second counting block responsive to a second group of timing signals providing a second count sequence for generating a second count signal upon completion of a stepping through said second count sequence;
- (c) a first delay block receiving said first count signal for generating a first delayed count signal;
- (d) a second delay block receiving said second count signal for generating a second delayed count signal;

- (e) a first selecting block receiving said first count signal and said first delayed count signal for selectively providing either the first count signal or the first delayed count signal as a first output count signal to a combining block;
- (f) a second selecting block receiving said second count signal and said second delayed count signal for selectively providing either the second count signal or the second delayed count signal as a second output count signal to said combining block; and
- (g) the combining block for combining the first and second output count signals and generating a feedback signal for enabling loading of the first and second counting blocks with new values for said first and second groups of timing signals.

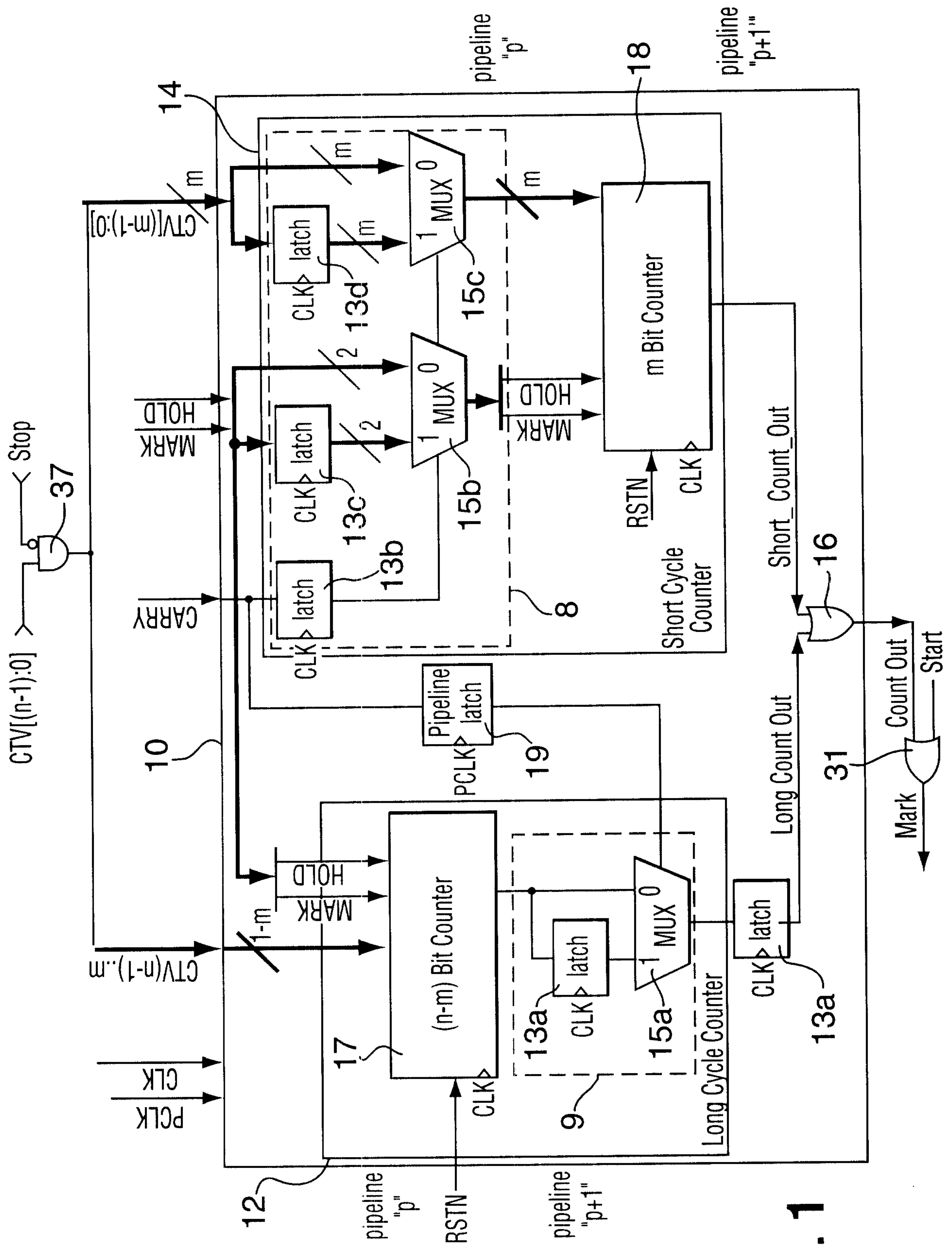


FIG. 1

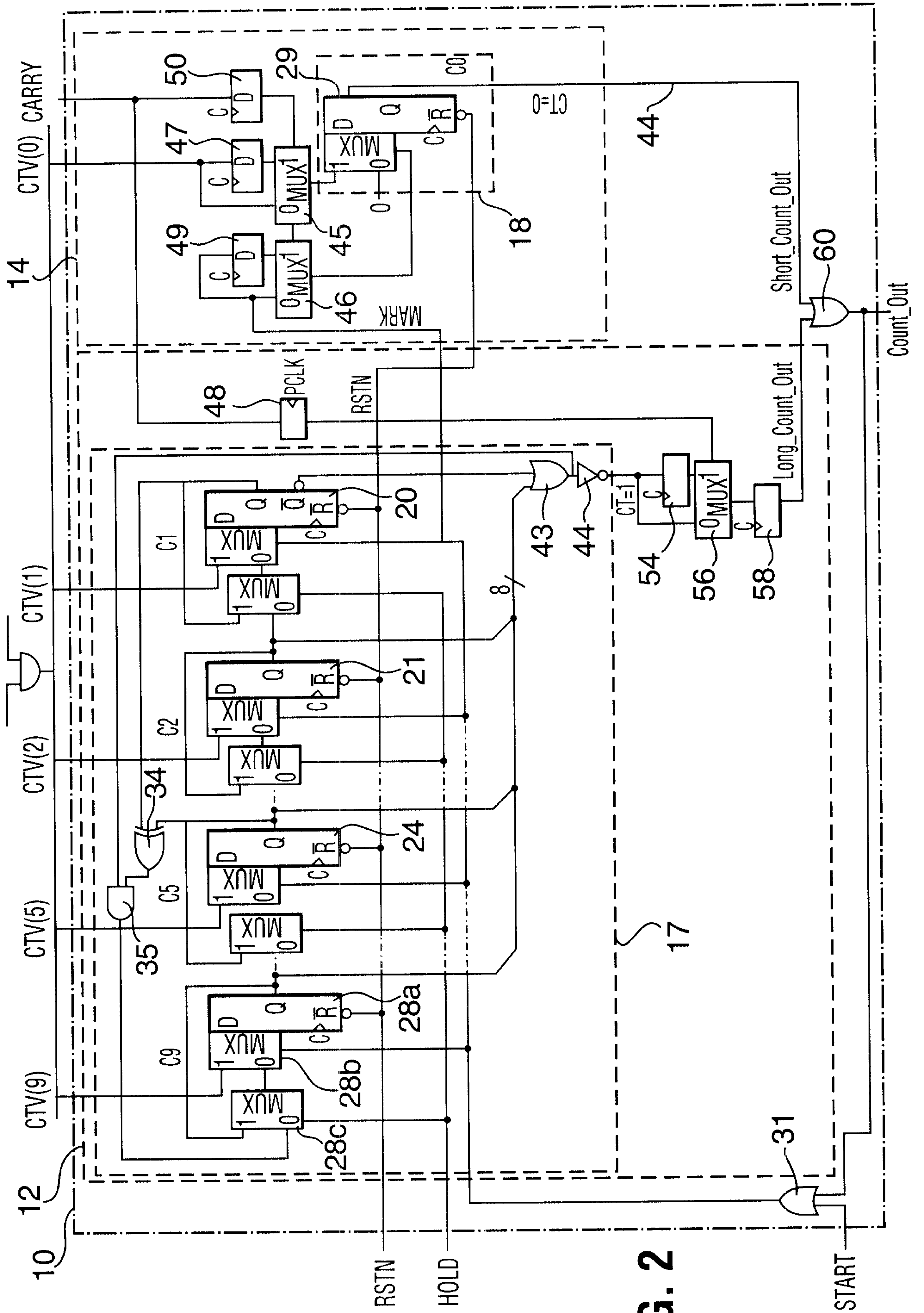


FIG. 2
Gowling Lafleur Henderson LLP

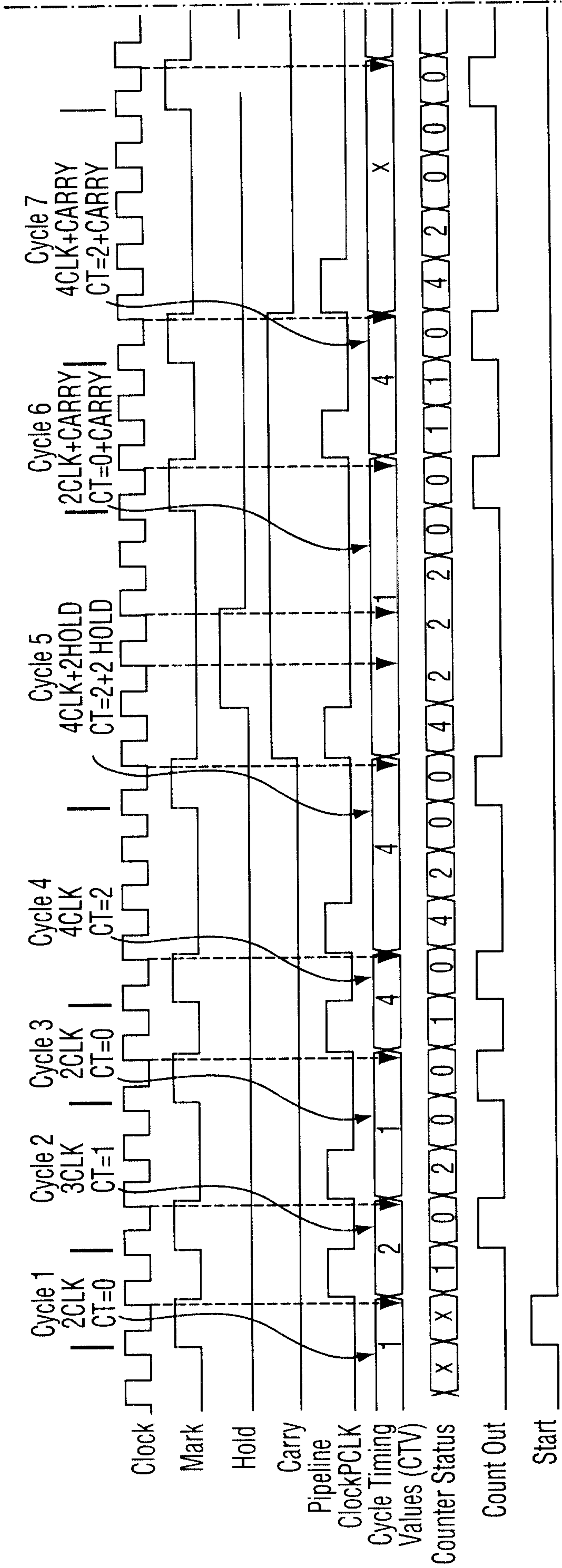


FIG. 3

FIG. 4

CTV(0)	CTV(1)	CTV(2)	CTV(3)	CTV(4)	CTV(5)	CTV(6)	CTV(7)	CTV(8)	CTV(9)	# clk	Edge Timing Value		Count#
											Decimal	Hex	
1	0	0	0	0	0	0	0	0	0	2	1	1	0
0	1	0	0	0	0	0	0	0	0	3	2	2	1
0	0	1	0	0	0	0	0	0	0	4	4	4	2
0	0	0	1	0	0	0	0	0	0	5	8	8	3
0	0	0	0	1	0	0	0	0	0	6	16	10	4
0	1	0	0	0	1	0	0	0	0	7	34	22	5
0	0	1	0	0	0	1	0	0	0	8	68	44	6
0	0	0	1	0	0	0	1	0	0	9	136	88	7
0	0	0	0	1	0	0	0	1	0	10	272	110	8
0	1	0	0	0	1	0	0	0	1	11	546	222	9
0	1	1	0	0	0	1	0	0	0	12	70	46	10
0	0	1	1	0	0	0	1	0	0	13	140	8C	11
0	0	0	1	1	0	0	0	1	0	14	280	118	12
0	1	0	0	1	1	0	0	0	1	15	562	232	13
0	0	1	0	0	1	1	0	0	0	16	100	64	14
0	0	0	1	0	0	1	1	0	0	17	200	C8	15
0	0	0	0	1	0	0	1	1	0	18	400	190	16
0	1	0	0	0	1	0	0	1	1	19	802	322	17
0	1	1	0	0	0	1	0	0	1	20	582	246	18
0	1	1	1	0	0	0	1	0	0	21	142	8E	19
0	0	1	1	1	0	0	0	1	0	22	284	11C	20
0	1	0	1	1	1	0	0	0	1	23	570	23A	21
0	0	1	0	1	1	1	0	0	0	24	116	74	22
0	1	0	1	0	1	1	1	0	0	25	234	EA	23
0	0	1	0	1	0	1	1	1	0	26	468	1D4	24
0	1	0	1	0	1	0	1	1	1	27	938	3AA	25
0	1	1	0	1	0	1	0	1	1	28	854	356	26
0	0	1	1	0	1	0	1	0	1	29	684	2AC	27
0	1	0	1	1	0	1	0	1	0	30	346	15A	28
0	1	1	0	1	1	0	1	0	1	31	694	2B6	29
0	0	1	1	0	1	1	0	1	0	32	364	16C	30
0	0	0	1	1	0	1	1	0	1	33	728	2D8	31
0	0	0	0	1	1	0	1	1	0	34	432	1B0	32
0	1	0	0	0	1	1	0	1	1	35	866	362	33
0	1	1	0	0	0	1	1	0	1	36	710	2C6	34
0	1	1	1	0	0	0	1	1	0	37	398	18E	35
0	0	1	1	1	0	0	0	1	1	38	796	31C	36
0	0	0	1	1	1	0	0	0	1	39	568	238	37

0	0	0	0	1	1	1	0	0	0	40	112	70	38
0	1	0	0	0	1	1	1	0	0	41	226	E2	39
0	0	1	0	0	0	1	1	1	0	42	452	1C4	40
0	0	0	1	0	0	0	1	1	1	43	904	388	41
0	1	0	0	1	0	0	0	1	1	44	786	312	42
0	0	1	0	0	1	0	0	0	1	45	548	224	43
0	1	0	1	0	0	1	0	0	0	46	74	4A	44
0	0	1	0	1	0	0	1	0	0	47	148	94	45
0	1	0	1	0	1	0	0	1	0	48	298	12A	46
0	0	1	0	1	0	1	0	0	1	49	596	254	47
0	0	0	1	0	1	0	1	0	0	50	168	A8	48
0	0	0	0	1	0	1	0	1	0	51	336	150	49
0	1	0	0	0	1	0	1	0	1	52	674	2A2	50
0	1	1	0	0	0	1	0	1	0	53	326	146	51
0	0	1	1	0	0	0	1	0	1	54	652	28C	52
0	1	0	1	1	0	0	0	1	0	55	282	11A	53
0	1	1	0	1	1	0	0	0	1	56	566	236	54
0	0	1	1	0	1	1	0	0	0	57	108	6C	55
0	0	0	1	1	0	1	1	0	0	58	216	D8	56
0	1	0	0	1	1	0	1	1	0	59	434	1B2	57
0	1	1	0	0	1	1	0	1	1	60	870	366	58
0	1	1	1	0	0	1	1	0	1	61	718	2CE	59
0	1	1	1	1	0	0	1	1	0	62	414	19E	60
0	1	1	1	1	1	0	0	1	1	63	830	33E	61
0	0	1	1	1	1	1	0	0	1	64	636	27C	62
0	0	0	1	1	1	1	1	0	0	65	248	F8	63
0	1	0	0	1	1	1	1	1	0	66	498	1F2	64
0	1	1	0	0	1	1	1	1	1	67	998	3E6	65
0	1	1	1	0	0	1	1	1	1	68	974	3CE	66
0	1	1	1	1	0	0	1	1	1	69	926	39E	67
0	0	1	1	1	1	0	0	1	1	70	828	33C	68
0	0	0	1	1	1	1	0	0	1	71	632	278	69
0	0	0	0	1	1	1	1	0	0	72	240	F0	70
0	1	0	0	0	1	1	1	1	0	73	482	1E2	71
0	0	1	0	0	0	1	1	1	1	74	964	3C4	72
0	1	0	1	0	0	0	1	1	1	75	906	38A	73
0	1	1	0	1	0	0	0	1	1	76	790	316	74
0	0	1	1	0	1	0	0	0	1	77	556	22C	75
0	1	0	1	1	0	1	0	0	0	78	90	5A	76
0	1	1	0	1	1	0	1	0	0	79	182	B6	77

0	1	1	1	0	1	1	0	1	0	80	366	16E	78
0	0	1	1	1	0	1	1	0	1	81	732	2DC	79
0	0	0	1	1	1	0	1	1	0	82	440	1B8	80
0	1	0	0	1	1	1	0	1	1	83	882	372	81
0	0	1	0	0	1	1	1	0	1	84	740	2E4	82
0	1	0	1	0	0	1	1	1	0	85	458	1CA	83
0	0	1	0	1	0	0	1	1	1	86	916	394	84
0	0	0	1	0	1	0	0	1	1	87	808	328	85
0	1	0	0	1	0	1	0	0	1	88	594	252	86
0	0	1	0	0	1	0	1	0	0	89	164	A4	87
0	0	0	1	0	0	1	0	1	0	90	328	148	88
0	0	0	0	1	0	0	1	0	1	91	656	290	89
0	0	0	0	0	1	0	0	1	0	92	288	120	90
0	0	0	0	0	0	1	0	0	1	93	576	240	91
0	1	0	0	0	0	0	1	0	0	94	130	82	92
0	0	1	0	0	0	0	0	1	0	95	260	104	93
0	0	0	1	0	0	0	0	0	1	96	520	208	94
0	1	0	0	1	0	0	0	0	0	97	18	12	95
0	1	1	0	0	1	0	0	0	0	98	38	26	96
0	0	1	1	0	0	1	0	0	0	99	76	4C	97
0	0	0	1	1	0	0	1	0	0	100	152	98	98
0	1	0	0	1	1	0	0	1	0	101	306	132	99
0	1	1	0	0	1	1	0	0	1	102	614	266	100
0	1	1	1	0	0	1	1	0	0	103	206	CE	101
0	0	1	1	1	0	0	1	1	0	104	412	19C	102
0	1	0	1	1	1	0	0	1	1	105	826	33A	103
0	0	1	0	1	1	1	0	0	1	106	628	274	104
0	0	0	1	0	1	1	1	0	0	107	232	E8	105
0	0	0	0	1	0	1	1	1	0	108	464	1D0	106
0	1	0	0	0	1	0	1	1	1	109	930	3A2	107
0	1	1	0	0	0	1	0	1	1	110	838	346	108
0	1	1	1	0	0	0	1	0	1	111	654	28E	109
0	1	1	1	1	0	0	0	1	0	112	286	11E	110
0	1	1	1	1	1	0	0	0	1	113	574	23E	111
0	0	1	1	1	1	1	0	0	0	114	124	7C	112
0	1	0	1	1	1	1	1	0	0	115	250	FA	113
0	1	1	0	1	1	1	1	1	0	116	502	1F6	114
0	1	1	1	0	1	1	1	1	1	117	1006	3EE	115
0	1	1	1	1	0	1	1	1	1	118	990	3DE	116
0	0	1	1	1	1	0	1	1	1	119	956	3BC	117

0	0	0	1	1	1	1	0	1	1	120	888	378	118
0	0	0	0	1	1	1	1	0	1	121	752	2F0	119
0	0	0	0	0	1	1	1	1	0	122	480	1E0	120
0	0	0	0	0	0	1	1	1	1	123	960	3C0	121
0	1	0	0	0	0	0	1	1	1	124	898	382	122
0	1	1	0	0	0	0	0	1	1	125	774	306	123
0	1	1	1	0	0	0	0	0	1	126	526	20E	124
0	1	1	1	1	0	0	0	0	0	127	30	1E	125
0	1	1	1	1	1	0	0	0	0	128	62	3E	126
0	1	1	1	1	1	1	0	0	0	129	126	7E	127
0	1	1	1	1	1	1	1	0	0	130	254	FE	128
0	1	1	1	1	1	1	1	1	0	131	510	1FE	129
0	1	1	1	1	1	1	1	1	1	132	1022	3FE	130
0	0	1	1	1	1	1	1	1	1	133	1020	3FC	131
0	0	0	1	1	1	1	1	1	1	134	1016	3F8	132
0	0	0	0	1	1	1	1	1	1	135	1008	3F0	133
0	0	0	0	0	1	1	1	1	1	136	992	3E0	134
0	1	0	0	0	0	1	1	1	1	137	962	3C2	135
0	1	1	0	0	0	0	1	1	1	138	902	386	136
0	1	1	1	0	0	0	0	1	1	139	782	30E	137
0	1	1	1	1	0	0	0	0	1	140	542	21E	138
0	0	1	1	1	1	0	0	0	0	141	60	3C	139
0	1	0	1	1	1	1	0	0	0	142	122	7A	140
0	1	1	0	1	1	1	1	0	0	143	246	F6	141
0	1	1	1	0	1	1	1	1	0	144	494	1EE	142
0	0	1	1	1	0	1	1	1	1	145	988	3DC	143
0	0	0	1	1	1	0	1	1	1	146	952	3B8	144
0	0	0	0	1	1	1	0	1	1	147	880	370	145
0	0	0	0	0	1	1	1	0	1	148	736	2E0	146
0	1	0	0	0	0	1	1	1	0	149	450	1C2	147
0	0	1	0	0	0	0	1	1	1	150	900	384	148
0	1	0	1	0	0	0	0	1	1	151	778	30A	149
0	1	1	0	1	0	0	0	0	1	152	534	216	150
0	0	1	1	0	1	0	0	0	0	153	44	2C	151
0	0	0	1	1	0	1	0	0	0	154	88	58	152
0	1	0	0	1	1	0	1	0	0	155	178	B2	153
0	1	1	0	0	1	1	0	1	0	156	358	166	154
0	0	1	1	0	0	1	1	0	1	157	716	2CC	155
0	1	0	1	1	0	0	1	1	0	158	410	19A	156
0	1	1	0	1	1	0	0	1	1	159	822	336	157

0	0	1	1	0	1	1	0	0	1	160	620	26C	158
0	1	0	1	1	0	1	1	0	0	161	218	DA	159
0	1	1	0	1	1	0	1	1	0	162	438	1B6	160
0	1	1	1	0	1	1	0	1	1	163	878	36E	161
0	1	1	1	1	0	1	1	0	1	164	734	2DE	162
0	0	1	1	1	1	0	1	1	0	165	444	1BC	163
0	1	0	1	1	1	1	0	1	1	166	890	37A	164
0	0	1	0	1	1	1	1	0	1	167	756	2F4	165
0	0	0	1	0	1	1	1	1	0	168	488	1E8	166
0	0	0	0	1	0	1	1	1	1	169	976	3D0	167
0	0	0	0	0	1	0	1	1	1	170	928	3A0	168
0	1	0	0	0	0	1	0	1	1	171	834	342	169
0	1	1	0	0	0	0	1	0	1	172	646	286	170
0	1	1	1	0	0	0	0	1	0	173	270	10E	171
0	0	1	1	1	0	0	0	0	1	174	540	21C	172
0	0	0	1	1	1	0	0	0	0	175	56	38	173
0	1	0	0	1	1	1	0	0	0	176	114	72	174
0	1	1	0	0	1	1	1	0	0	177	230	E6	175
0	0	1	1	0	0	1	1	1	0	178	460	1CC	176
0	0	0	1	1	0	0	1	1	1	179	920	398	177
0	0	0	0	1	1	0	0	1	1	180	816	330	178
0	0	0	0	0	1	1	0	0	1	181	608	260	179
0	1	0	0	0	0	1	1	0	0	182	194	C2	180
0	0	1	0	0	0	0	1	1	0	183	388	184	181
0	0	0	1	0	0	0	0	1	1	184	776	308	182
0	1	0	0	1	0	0	0	0	1	185	530	212	183
0	0	1	0	0	1	0	0	0	0	186	36	24	184
0	0	0	1	0	0	1	0	0	0	187	72	48	185
0	0	0	0	1	0	0	1	0	0	188	144	90	186
0	1	0	0	0	1	0	0	1	0	189	290	122	187
0	0	1	0	0	0	1	0	0	1	190	580	244	188
0	1	0	1	0	0	0	1	0	0	191	138	8A	189
0	0	1	0	1	0	0	0	1	0	192	276	114	190
0	1	0	1	0	1	0	0	0	1	193	554	22A	191
0	1	1	0	1	0	1	0	0	0	194	86	56	192
0	1	1	1	0	1	0	1	0	0	195	174	AE	193
0	0	1	1	1	0	1	0	1	0	196	348	15C	194
0	1	0	1	1	1	0	1	0	1	197	698	2BA	195
0	0	1	0	1	1	1	0	1	0	198	372	174	196
0	1	0	1	0	1	1	1	0	1	199	746	2EA	197

0	1	1	0	1	0	1	1	1	0	200	470	1D6	198
0	1	1	1	0	1	0	1	1	1	201	942	3AE	199
0	1	1	1	1	0	1	0	1	1	202	862	35E	200
0	0	1	1	1	1	0	1	0	1	203	700	2BC	201
0	0	0	1	1	1	1	0	1	0	204	376	178	202
0	1	0	0	1	1	1	1	0	1	205	754	2F2	203
0	0	1	0	0	1	1	1	1	0	206	484	1E4	204
0	0	0	1	0	0	1	1	1	1	207	968	3C8	205
0	1	0	0	1	0	0	1	1	1	208	914	392	206
0	0	1	0	0	1	0	0	1	1	209	804	324	207
0	1	0	1	0	0	1	0	0	1	210	586	24A	208
0	1	1	0	1	0	0	1	0	0	211	150	96	209
0	1	1	1	0	1	0	0	1	0	212	302	12E	210
0	0	1	1	1	0	1	0	0	1	213	604	25C	211
0	0	0	1	1	1	0	1	0	0	214	184	B8	212
0	1	0	0	1	1	1	0	1	0	215	370	172	213
0	1	1	0	0	1	1	1	0	1	216	742	2E6	214
0	1	1	1	0	0	1	1	1	0	217	462	1CE	215
0	0	1	1	1	0	0	1	1	1	218	924	39C	216
0	0	0	1	1	1	0	0	1	1	219	824	338	217
0	0	0	0	1	1	1	0	0	1	220	624	270	218
0	0	0	0	0	1	1	1	0	0	221	224	E0	219
0	0	0	0	0	0	1	1	1	0	222	448	1C0	220
0	0	0	0	0	0	0	1	1	1	223	896	380	221
0	1	0	0	0	0	0	0	1	1	224	770	302	222
0	1	1	0	0	0	0	0	0	1	225	518	206	223
0	1	1	1	0	0	0	0	0	0	226	14	E	224
0	0	1	1	1	0	0	0	0	0	227	28	1C	225
0	1	0	1	1	1	0	0	0	0	228	58	3A	226
0	1	1	0	1	1	1	0	0	0	229	118	76	227
0	1	1	1	0	1	1	1	0	0	230	238	EE	228
0	0	1	1	1	0	1	1	1	0	231	476	1DC	229
0	1	0	1	1	1	0	1	1	1	232	954	3BA	230
0	0	1	0	1	1	1	0	1	1	233	884	374	231
0	0	0	1	0	1	1	1	0	1	234	744	2E8	232
0	1	0	0	1	0	1	1	1	0	235	466	1D2	233
0	1	1	0	0	1	0	1	1	1	236	934	3A6	234
0	1	1	1	0	0	1	0	1	1	237	846	34E	235
0	1	1	1	1	0	0	1	0	1	238	670	29E	236
0	0	1	1	1	1	0	0	1	0	239	316	13C	237

0	1	0	1	1	1	1	0	0	1	240	634	27A	238
0	0	1	0	1	1	1	1	0	0	241	244	F4	239
0	1	0	1	0	1	1	1	1	0	242	490	1EA	240
0	0	1	0	1	0	1	1	1	1	243	980	3D4	241
0	0	0	1	0	1	0	1	1	1	244	936	3A8	242
0	1	0	0	1	0	1	0	1	1	245	850	352	243
0	0	1	0	0	1	0	1	0	1	246	676	2A4	244
0	1	0	1	0	0	1	0	1	0	247	330	14A	245
0	0	1	0	1	0	0	1	0	1	248	660	294	246
0	0	0	1	0	1	0	0	1	0	249	296	128	247
0	0	0	0	1	0	1	0	0	1	250	592	250	248
0	0	0	0	0	1	0	1	0	0	251	160	A0	249
0	0	0	0	0	0	1	0	1	0	252	320	140	250
0	0	0	0	0	0	0	1	0	1	253	640	280	251
0	1	0	0	0	0	0	0	1	0	254	258	102	252
0	0	1	0	0	0	0	0	0	1	255	516	204	253
0	1	0	1	0	0	0	0	0	0	256	10	A	254
0	0	1	0	1	0	0	0	0	0	257	20	14	255
0	1	0	1	0	1	0	0	0	0	258	42	2A	256
0	0	1	0	1	0	1	0	0	0	259	84	54	257
0	1	0	1	0	1	0	1	0	0	260	170	AA	258
0	0	1	0	1	0	1	0	1	0	261	340	154	259
0	1	0	1	0	1	0	1	0	1	262	682	2AA	260
0	1	1	0	1	0	1	0	1	0	263	342	156	261
0	1	1	1	0	1	0	1	0	1	264	686	2AE	262
0	1	1	1	1	0	1	0	1	0	265	350	15E	263
0	1	1	1	1	1	0	1	0	1	266	702	2BE	264
0	0	1	1	1	1	1	0	1	0	267	380	17C	265
0	1	0	1	1	1	1	1	0	1	268	762	2FA	266
0	0	1	0	1	1	1	1	1	0	269	500	1F4	267
0	1	0	1	0	1	1	1	1	1	270	1002	3EA	268
0	1	1	0	1	0	1	1	1	1	271	982	3D6	269
0	0	1	1	0	1	0	1	1	1	272	940	3AC	270
0	1	0	1	1	0	1	0	1	1	273	858	35A	271
0	0	1	0	1	1	0	1	0	1	274	692	2B4	272
0	0	0	1	0	1	1	0	1	0	275	360	168	273
0	0	0	0	1	0	1	1	0	1	276	720	2D0	274
0	0	0	0	0	1	0	1	1	0	277	416	1A0	275
0	0	0	0	0	0	1	0	1	1	278	832	340	276
0	1	0	0	0	0	0	1	0	1	279	642	282	277

0	1	1	0	0	0	0	0	1	0	280	262	106	278
0	0	1	1	0	0	0	0	0	1	281	524	20C	279
0	1	0	1	1	0	0	0	0	0	282	26	1A	280
0	1	1	0	1	1	0	0	0	0	283	54	36	281
0	1	1	1	0	1	1	0	0	0	284	110	6E	282
0	0	1	1	1	0	1	1	0	0	285	220	DC	283
0	1	0	1	1	1	0	1	1	0	286	442	1BA	284
0	1	1	0	1	1	1	0	1	1	287	886	376	285
0	0	1	1	0	1	1	1	0	1	288	748	2EC	286
0	1	0	1	1	0	1	1	1	0	289	474	1DA	287
0	1	1	0	1	1	0	1	1	1	290	950	3B6	288
0	0	1	1	0	1	1	0	1	1	291	876	36C	289
0	1	0	1	1	0	1	1	0	1	292	730	2DA	290
0	0	1	0	1	1	0	1	1	0	293	436	1B4	291
0	1	0	1	0	1	1	0	1	1	294	874	36A	292
0	1	1	0	1	0	1	1	0	1	295	726	2D6	293
0	0	1	1	0	1	0	1	1	0	296	428	1AC	294
0	0	0	1	1	0	1	0	1	1	297	856	358	295
0	0	0	0	1	1	0	1	0	1	298	688	2B0	296
0	0	0	0	0	1	1	0	1	0	299	352	160	297
0	0	0	0	0	0	1	1	0	1	300	704	2C0	298
0	1	0	0	0	0	0	1	1	0	301	386	182	299
0	0	1	0	0	0	0	0	1	1	302	772	304	300
0	1	0	1	0	0	0	0	0	1	303	522	20A	301
0	1	1	0	1	0	0	0	0	0	304	22	16	302
0	1	1	1	0	1	0	0	0	0	305	46	2E	303
0	0	1	1	1	0	1	0	0	0	306	92	5C	304
0	1	0	1	1	1	0	1	0	0	307	186	BA	305
0	1	1	0	1	1	1	0	1	0	308	374	176	306
0	1	1	1	0	1	1	1	0	1	309	750	2EE	307
0	1	1	1	1	0	1	1	1	0	310	478	1DE	308
0	1	1	1	1	1	0	1	1	1	311	958	3BE	309
0	0	1	1	1	1	1	0	1	1	312	892	37C	310
0	0	0	1	1	1	1	1	0	1	313	760	2F8	311
0	0	0	0	1	1	1	1	1	0	314	496	1F0	312
0	1	0	0	0	1	1	1	1	1	315	994	3E2	313
0	1	1	0	0	0	1	1	1	1	316	966	3C6	314
0	1	1	1	0	0	0	1	1	1	317	910	38E	315
0	1	1	1	1	0	0	0	1	1	318	798	31E	316
0	0	1	1	1	1	0	0	0	1	319	572	23C	317

0	0	0	1	1	1	1	0	0	0	320	120	78	318
0	1	0	0	1	1	1	1	0	0	321	242	F2	319
0	1	1	0	0	1	1	1	1	0	322	486	1E6	320
0	0	1	1	0	0	1	1	1	1	323	972	3CC	321
0	1	0	1	1	0	0	1	1	1	324	922	39A	322
0	0	1	0	1	1	0	0	1	1	325	820	334	323
0	0	0	1	0	1	1	0	0	1	326	616	268	324
0	1	0	0	1	0	1	1	0	0	327	210	D2	325
0	1	1	0	0	1	0	1	1	0	328	422	1A6	326
0	0	1	1	0	0	1	0	1	1	329	844	34C	327
0	1	0	1	1	0	0	1	0	1	330	666	29A	328
0	0	1	0	1	1	0	0	1	0	331	308	134	329
0	1	0	1	0	1	1	0	0	1	332	618	26A	330
0	1	1	0	1	0	1	1	0	0	333	214	D6	331
0	1	1	1	0	1	0	1	1	0	334	430	1AE	332
0	0	1	1	1	0	1	0	1	1	335	860	35C	333
0	0	0	1	1	1	0	1	0	1	336	696	2B8	334
0	0	0	0	1	1	1	0	1	0	337	368	170	335
0	1	0	0	0	1	1	1	0	1	338	738	2E2	336
0	1	1	0	0	0	1	1	1	0	339	454	1C6	337
0	0	1	1	0	0	0	1	1	1	340	908	38C	338
0	1	0	1	1	0	0	0	1	1	341	794	31A	339
0	0	1	0	1	1	0	0	0	1	342	564	234	340
0	0	0	1	0	1	1	0	0	0	343	104	68	341
0	0	0	0	1	0	1	1	0	0	344	208	D0	342
0	1	0	0	0	1	0	1	1	0	345	418	1A2	343
0	0	1	0	0	0	1	0	1	1	346	836	344	344
0	1	0	1	0	0	0	1	0	1	347	650	28A	345
0	1	1	0	1	0	0	0	1	0	348	278	116	346
0	1	1	1	0	1	0	0	0	1	349	558	22E	347
0	1	1	1	1	0	1	0	0	0	350	94	5E	348
0	1	1	1	1	1	0	1	0	0	351	190	BE	349
0	1	1	1	1	1	1	0	1	0	352	382	17E	350
0	1	1	1	1	1	1	1	0	1	353	766	2FE	351
0	0	1	1	1	1	1	1	1	0	354	508	1FC	352
0	1	0	1	1	1	1	1	1	1	355	1018	3FA	353
0	0	1	0	1	1	1	1	1	1	356	1012	3F4	354
0	0	0	1	0	1	1	1	1	1	357	1000	3E8	355
0	1	0	0	1	0	1	1	1	1	358	978	3D2	356
0	0	1	0	0	1	0	1	1	1	359	932	3A4	357

0	1	0	1	0	0	1	0	1	1	360	842	34A	358
0	1	1	0	1	0	0	1	0	1	361	662	296	359
0	0	1	1	0	1	0	0	1	0	362	300	12C	360
0	0	0	1	1	0	1	0	0	1	363	600	258	361
0	0	0	0	1	1	0	1	0	0	364	176	B0	362
0	1	0	0	0	1	1	0	1	0	365	354	162	363
0	0	1	0	0	0	1	1	0	1	366	708	2C4	364
0	1	0	1	0	0	0	1	1	0	367	394	18A	365
0	0	1	0	1	0	0	0	1	1	368	788	314	366
0	0	0	1	0	1	0	0	0	1	369	552	228	367
0	1	0	0	1	0	1	0	0	0	370	82	52	368
0	1	1	0	0	1	0	1	0	0	371	166	A6	369
0	0	1	1	0	0	1	0	1	0	372	332	14C	370
0	0	0	1	1	0	0	1	0	1	373	664	298	371
0	0	0	0	1	1	0	0	1	0	374	304	130	372
0	1	0	0	0	1	1	0	0	1	375	610	262	373
0	1	1	0	0	0	1	1	0	0	376	198	C6	374
0	0	1	1	0	0	0	1	1	0	377	396	18C	375
0	0	0	1	1	0	0	0	1	1	378	792	318	376
0	0	0	0	1	1	0	0	0	1	379	560	230	377
0	0	0	0	0	1	1	0	0	0	380	96	60	378
0	0	0	0	0	0	1	1	0	0	381	192	C0	379
0	0	0	0	0	0	0	1	1	0	382	384	180	380
0	0	0	0	0	0	0	0	1	1	383	768	300	381
0	1	0	0	0	0	0	0	0	1	384	514	202	382
0	1	1	0	0	0	0	0	0	0	385	6	6	383
0	0	1	1	0	0	0	0	0	0	386	12	C	384
0	0	0	1	1	0	0	0	0	0	387	24	18	385
0	1	0	0	1	1	0	0	0	0	388	50	32	386
0	1	1	0	0	1	1	0	0	0	389	102	66	387
0	0	1	1	0	0	1	1	0	0	390	204	CC	388
0	0	0	1	1	0	0	1	1	0	391	408	198	389
0	1	0	0	1	1	0	0	1	1	392	818	332	390
0	0	1	0	0	1	1	0	0	1	393	612	264	391
0	1	0	1	0	0	1	1	0	0	394	202	CA	392
0	0	1	0	1	0	0	1	1	0	395	404	194	393
0	1	0	1	0	1	0	0	1	1	396	810	32A	394
0	1	1	0	1	0	1	0	0	1	397	598	256	395
0	0	1	1	0	1	0	1	0	0	398	172	AC	396
0	0	0	1	1	0	1	0	1	0	399	344	158	397

0	1	0	0	1	1	0	1	0	1	400	690	2B2	398
0	0	1	0	0	1	1	0	1	0	401	356	164	399
0	0	0	1	0	0	1	1	0	1	402	712	2C8	400
0	1	0	0	1	0	0	1	1	0	403	402	192	401
0	1	1	0	0	1	0	0	1	1	404	806	326	402
0	1	1	1	0	0	1	0	0	1	405	590	24E	403
0	1	1	1	1	0	0	1	0	0	406	158	9E	404
0	1	1	1	1	1	0	0	1	0	407	318	13E	405
0	1	1	1	1	1	1	0	0	1	408	638	27E	406
0	0	1	1	1	1	1	1	0	0	409	252	FC	407
0	1	0	1	1	1	1	1	1	0	410	506	1FA	408
0	1	1	0	1	1	1	1	1	1	411	1014	3F6	409
0	0	1	1	0	1	1	1	1	1	412	1004	3EC	410
0	1	0	1	1	0	1	1	1	1	413	986	3DA	411
0	0	1	0	1	1	0	1	1	1	414	948	3B4	412
0	0	0	1	0	1	1	0	1	1	415	872	368	413
0	1	0	0	1	0	1	1	0	1	416	722	2D2	414
0	0	1	0	0	1	0	1	1	0	417	420	1A4	415
0	0	0	1	0	0	1	0	1	1	418	840	348	416
0	1	0	0	1	0	0	1	0	1	419	658	292	417
0	0	1	0	0	1	0	0	1	0	420	292	124	418
0	0	0	1	0	0	1	0	0	1	421	584	248	419
0	1	0	0	1	0	0	1	0	0	422	146	92	420
0	1	1	0	0	1	0	0	1	0	423	294	126	421
0	0	1	1	0	0	1	0	0	1	424	588	24C	422
0	1	0	1	1	0	0	1	0	0	425	154	9A	423
0	1	1	0	1	1	0	0	1	0	426	310	136	424
0	1	1	1	0	1	1	0	0	1	427	622	26E	425
0	1	1	1	1	0	1	1	0	0	428	222	DE	426
0	1	1	1	1	1	0	1	1	0	429	446	1BE	427
0	1	1	1	1	1	1	0	1	1	430	894	37E	428
0	0	1	1	1	1	1	1	0	1	431	764	2FC	429
0	0	0	1	1	1	1	1	1	0	432	504	1F8	430
0	1	0	0	1	1	1	1	1	1	433	1010	3F2	431
0	0	1	0	0	1	1	1	1	1	434	996	3E4	432
0	1	0	1	0	0	1	1	1	1	435	970	3CA	433
0	1	1	0	1	0	0	1	1	1	436	918	396	434
0	0	1	1	0	1	0	0	1	1	437	812	32C	435
0	1	0	1	1	0	1	0	0	1	438	602	25A	436
0	0	1	0	1	1	0	1	0	0	439	180	B4	437

0	1	0	1	0	1	1	0	1	0	440	362	16A	438
0	0	1	0	1	0	1	1	0	1	441	724	2D4	439
0	0	0	1	0	1	0	1	1	0	442	424	1A8	440
0	0	0	0	1	0	1	0	1	1	443	848	350	441
0	0	0	0	0	1	0	1	0	1	444	672	2A0	442
0	1	0	0	0	0	1	0	1	0	445	322	142	443
0	0	1	0	0	0	0	1	0	1	446	644	284	444
0	1	0	1	0	0	0	0	1	0	447	266	10A	445
0	0	1	0	1	0	0	0	0	1	448	532	214	446
0	0	0	1	0	1	0	0	0	0	449	40	28	447
0	0	0	0	1	0	1	0	0	0	450	80	50	448
0	1	0	0	0	1	0	1	0	0	451	162	A2	449
0	0	1	0	0	0	1	0	1	0	452	324	144	450
0	0	0	1	0	0	0	1	0	1	453	648	288	451
0	1	0	0	1	0	0	0	1	0	454	274	112	452
0	1	1	0	0	1	0	0	0	1	455	550	226	453
0	1	1	1	0	0	1	0	0	0	456	78	4E	454
0	0	1	1	1	0	0	1	0	0	457	156	9C	455
0	1	0	1	1	1	0	0	1	0	458	314	13A	456
0	1	1	0	1	1	1	1	0	0	459	630	276	457
0	0	1	1	0	1	1	1	1	0	460	236	EC	458
0	0	0	1	1	0	1	1	1	0	461	472	1D8	459
0	1	0	0	1	1	0	1	1	1	462	946	3B2	460
0	0	1	0	0	1	1	0	1	1	463	868	364	461
0	1	0	1	0	0	1	1	0	1	464	714	2CA	462
0	1	1	0	1	0	0	1	1	0	465	406	196	463
0	1	1	1	0	1	0	0	1	1	466	814	32E	464
0	1	1	1	1	0	1	0	0	1	467	606	25E	465
0	0	1	1	1	1	0	1	0	0	468	188	BC	466
0	1	0	1	1	1	1	0	1	0	469	378	17A	467
0	1	1	0	1	1	1	1	0	1	470	758	2F6	468
0	0	1	1	0	1	1	1	1	0	471	492	1EC	469
0	0	0	1	1	0	1	1	1	1	472	984	3D8	470
0	0	0	0	1	1	0	1	1	1	473	944	3B0	471
0	0	0	0	0	1	1	0	1	1	474	864	360	472
0	1	0	0	0	0	1	1	0	1	475	706	2C2	473
0	1	1	0	0	0	0	1	1	0	476	390	186	474
0	0	1	1	0	0	0	0	1	1	477	780	30C	475
0	1	0	1	1	0	0	0	0	1	478	538	21A	476
0	0	1	0	1	1	0	0	0	0	479	52	34	477

0	1	0	1	0	1	1	0	0	0	480	106	6A	478
0	0	1	0	1	0	1	1	0	0	481	212	D4	479
0	1	0	1	0	1	0	1	1	0	482	426	1AA	480
0	0	1	0	1	0	1	0	1	1	483	852	354	481
0	0	0	1	0	1	0	1	0	1	484	680	2A8	482
0	1	0	0	1	0	1	0	1	0	485	338	152	483
0	1	1	0	0	1	0	1	0	1	486	678	2A6	484
0	1	1	1	0	0	1	0	1	0	487	334	14E	485
0	0	1	1	1	0	0	1	0	1	488	668	29C	486
0	0	0	1	1	1	0	0	1	0	489	312	138	487
0	1	0	0	1	1	1	0	0	1	490	626	272	488
0	0	1	0	0	1	1	1	0	0	491	228	E4	489
0	0	0	1	0	0	1	1	1	0	492	456	1C8	490
0	0	0	0	1	0	0	1	1	1	493	912	390	491
0	0	0	0	0	1	0	0	1	1	494	800	320	492
0	1	0	0	0	0	1	0	0	1	495	578	242	493
0	1	1	0	0	0	0	1	0	0	496	134	86	494
0	0	1	1	0	0	0	0	1	0	497	268	10C	495
0	0	0	1	1	0	0	0	0	1	498	536	218	496
0	0	0	0	1	1	0	0	0	0	499	48	30	497
0	1	0	0	0	1	1	0	0	0	500	98	62	498
0	0	1	0	0	0	1	1	0	0	501	196	C4	499
0	0	0	1	0	0	0	1	1	0	502	392	188	500
0	0	0	0	1	0	0	0	1	1	503	784	310	501
0	0	0	0	0	1	0	0	0	1	504	544	220	502
0	1	0	0	0	0	1	0	0	0	505	66	42	503
0	0	1	0	0	0	0	1	0	0	506	132	84	504
0	0	0	1	0	0	0	0	1	0	507	264	108	505
0	0	0	0	1	0	0	0	0	1	508	528	210	506
0	0	0	0	0	1	0	0	0	0	509	32	20	507
0	0	0	0	0	0	1	0	0	0	510	64	40	508
0	0	0	0	0	0	0	1	0	0	511	128	80	509
0	0	0	0	0	0	0	0	1	0	512	256	100	510
0	0	0	0	0	0	0	0	0	1	513	512	200	511

