

PROGRAMMABLE DIVIDER

CROSS REFERENCE TO RELATED DOCUMENTS

This application is related to U.S. Patent Application Serial No. 09/637,534 for a "Gate Transition Counter", a copy of which is file with the application as Annex A.

FIELD OF THE INVENTION

This invention relates generally to the field of circuits and methods for performing binary math. More particularly, in certain embodiments, this invention relates to a special purpose programmable divider circuit.

BACKGROUND OF THE INVENTION

Full arithmetic logic units (ALU) are commonly available and are used to provide general-purpose mathematical functions including add, subtract, multiply and divide. While such arithmetic logic units are versatile devices, the full power of such a device is not always required for a specific application. By way of example, if the only function required is an integer divide function where the remainder is discarded, a full arithmetic logic unit poses a number of disadvantages. Use of a full arithmetic logic unit in such an application would increase the expense of a circuit as well as require

substantial amounts of silicon to implement. Moreover, the general-purpose nature of such device can lead to compromises that inhibit performance. Also, many if not all, arithmetic logic units are clocked devices, and thus require support circuitry and clocking. This can further add to expense and further compromise in performance.

In certain applications, therefore, it is advantageous to provide a special purpose circuit to provide a mathematical function.

BRIEF SUMMARY OF THE INVENTION

The present invention relates generally to a programmable divider circuit and methods therefor. Objects, advantages and features of the invention will become apparent to those skilled in the art upon consideration of the following detailed description of the invention.

In one embodiment of a programmable divider, consistent with the present invention, for dividing an N bit binary number F by a programmed value equal to $(2^{N-1}) / K$, where K is a selected integer between 1 and $(2^{N-1} - 1)$ to produce a divided binary number P, N -1 shift circuits divide the N bit binary number F by shifting the N bit binary number F to the right by 1 through (N-1) bits respectively. The shift circuits produce N-1 divided values G_1 through G_{N-1} having values of $F/2$ through $F / (2^{N-1})$ respectively. A plurality of adders adds any selected combination of the divided values G_1 through G_{N-1} together to produce the divided binary number P.

A method, consistent with the present invention, of dividing an N bit binary number F by a programmed value equal to $(2^{N-1}) / K$, where K is a

selected integer between 1 and $(2^{N-1} - 1)$, to produce a divided binary number P, includes: shifting the N bit binary number F to the right by 1 through (N-1) bits respectively to produce N-1 divided values, G_1 through G_{N-1} , having values of $F/2$ through $F / (2^{N-1})$ respectively; and adding any selected combination of the divided values, G_1 through G_{N-1} , together to produce the divided binary number P.

In another embodiment a programmable divider consistent with the present invention, for dividing an N bit binary number F by a programmed value equal to $(2^{N-1}) / K$, where K is a selected integer between 1 and $(2^{N-1} - 1)$ to produce a divided binary number P, includes N -1 shift circuits for dividing the N bit binary number F by shifting the N bit binary number F to the right by 1 through (N-1) bits respectively, wherein the shift circuits produce N-1 divided values, G_1 through G_{N-1} , having values of $F/2$ through $F / (2^{N-1})$ respectively. A plurality of one bit full adders with carry is arranged in N-2 arrays of N one bit full adders with carry to add any selected combination of the divided values G_1 through G_{N-1} or zero together to produce the divided binary number P. A plurality of switches, each comprising a two input multiplexer, is arranged in N-1 arrays that selectively couple either a divided value G_1 through G_{N-1} or zero to an input of each adder.

Many variations, equivalents and permutations of the above illustrative exemplary embodiments of the invention will occur to those skilled in the art upon consideration of the description that follows. The particular examples above should not be considered to define the scope of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

The features of the invention believed to be novel are set forth with particularity in the appended claims. The invention itself however, both as to organization and method of operation, together with objects and advantages thereof, may be best understood by reference to the following detailed description of the invention, which describes certain exemplary embodiments of the invention, taken in conjunction with the accompanying drawings in which:

FIG. 1 is a schematic drawing of an embodiment of a programmable divider, consistent with the present invention, which selectively divides a seven bit binary number.

FIG. 2 is a schematic of a one bit full adder with carry suitable for use in the present invention.

DETAILED DESCRIPTION OF THE INVENTION

While this invention is susceptible of embodiment in many different forms, there is shown in the drawings and will herein be described in detail specific embodiments, with the understanding that the present disclosure is to be considered as an example of the principles of the invention and not intended to limit the invention to the specific embodiments shown and described. In the description below, like reference numerals are used to describe the same, similar or corresponding parts in the several views of the drawings.

Referring now to **FIG. 1**, a programmable divider circuit in accordance with one embodiment of the present invention is illustrated. The programmable divider circuit illustrated receives an input binary number F , which in this case is a seven bit binary number having bits f_6 through f_0 , with f_0 being the least significant bit. Circuit 10 produces an output binary number P having bits p_6 through p_0 with p_0 being the least significant bit. Six programming inputs 12, 14, 16, 18, 20 and 22 are provided to selectively divide the value of F by any selected divisor between 2 and 64 in increments of 2^J , where J is an integer in this case between one and six. Of course, those skilled in the art will recognize that the topology illustrated can be expanded or reduced to any size binary number and is not restricted to a seven bit binary number. In general, an N bit binary number can be accommodated.

In divider 10, if input 12 is selected, the binary input F is divided by 2 ($F/2=G_1$). If input 14 is selected, the binary number F is divided by 4 ($F/4=G_2$). If both inputs 12 and 14 are selected, the binary number F is divided 2 and by 4, and the results are added to produce division by $4/3$ ($=G_1 + G_2 =$ multiplication by $3/4$). If input 16 is selected, the binary number F is divided by 8 ($F/8= G_3$). In a manner similar to that described above, if all three input lines 12, 14 and 16 are selected, the binary number F is divided by 2, 4 and 8. These divided values of F are then added together and the resulting output is $F(7/8) = F/2 + F/4 + F/8 = G_1 + G_2 + G_3$. Input lines 12, 14, 16, 18, 20 and 22 can be thus selected in any combination to achieve the desired divisor. In a manner similar to that described above, selection of line 18 results in division of F by 16 ($F/16 = G_4$), selection of line 20 results in division of F by 32 ($F/32 = G_5$) and selection of line 22 results in division of F by 64 ($F/64 = G_6$).

Each of lines 12, 14, 16, 18, 20 and 22 control one or more multiplexers (or switches). By way of example, line 12 controls an array of six multiplexers 26 made up of multiplexers 28, 30, 32, 34, 36 and 38. Multiplexers 28, 30, 32, 34, 36 and 38 are two input switching devices controlled by line 12 with the logic value at line 12 determining which of the two inputs is passed to an output. Six such multiplexers are used in multiplexer array 26 to selectively apply the value of F shifted to the right by one bit (G_1) to an array of binary full adders 44. If the line 12 is selected, the values f_1 through f_6 (representing $G_1 = F$ shifted to the right by one bit = $F/2$) are applied to one input of the array of adders 44 with the most significant bits of the seven bit word backfilled with zeros. If line 12 is not selected, logic zeros are applied to each of the adders in the six least significant bits of adder array 44 (i.e. $G_1 = 0$). Thus, by selection of line 12, a contribution to the value of P equaling F divided by 2 is added by the divider circuit 10.

Adder array 44 is made up of seven adders (one for each bit of the output number P) shown as adders 50, 52, 53, 54, 56, 58 and 60. Adder 60 produces an output corresponding to a least significant bit where adder 50 produces an output corresponding to a most significant bit. Each adder in array 44 is a one bit two input full adder with carry. Each carry output is applied as a carry input to the adder at the next more significant bit location. Thus, the carry output of adder 60 is applied to the carry input of adder 58, the carry output of adder 58 is applied to the carry input of adder 56, and so on. The outputs of each of the six multiplexers of multiplexer array 26 is applied to one of the inputs (input B is shown) for the six least significant bit locations in adder array 44. Thus, the output of multiplexers 38 is applied to the B input of adder 60, the output of multiplexer 36 is applied to the B input

of multiplexer 58, the output of multiplexer 34 is applied to the B input of adder 56, and so on. Adder 50 serves to receive the carry output of adder 52 with both other inputs tied to logic zero.

In a similar manner, line 14 controls an array 66 of 5 multiplexers which, depending upon the value applied to line 14, selectively adds either zero or F shifted to the right by two bits (G_2) to the least significant bit adders of adder array 44. Thus, multiplexer array 66 is made up of five multiplexers, 70, 72, 74, 76 and 78 with inputs f_2, f_3, f_4, f_5 and f_6 respectively applied to one input of each multiplexer and zero applied to the other input of each multiplexer. The output of multiplexer 70 is applied to the A input of adder 60, the output of multiplexer 72 is applied to the A input of adder 58 and so on until the output of multiplexer 78 is applied to the A input of adder 53. The A inputs of adders 50 and 52 are both connected to zero's.

As illustrated, the adders 44 produce a seven bit binary number that can selectively add $f/2$ to $f/4$ at output terminals S of the array of adders 44. Line 16 controls an array 82 of four multiplexers which selectively add the value of F shifted to the right by three bits to produce $F/8 = G_3$. This value (backfilled with zeros at the most significant bits) is applied to the A input of a second array 84 of one bit binary full adders arranged identically to array 44. This second array of full adders receives the output of the third array of multiplexers (representing $F/8$) at the A input and receives the output of the array of adders 44 at the B inputs thereof. Thus, the output of the second array of adders 84 is a seven bit binary number which represents a value equaling the sum of any selected ones of $f/2, f/4$ and $f/8$ (G_1, G_2 and G_3) as desired by selection of lines 12, 14 and 16.

In a like manner, line 18 controls an array of three multiplexers 86 which selectively add either zero's or F shifted by four bits (G_4) to the right to the output of the second array of adders 84 in a third array of adders 88. Line 20 controls an array of two multiplexers 92, which selectively add either zero's or F shifted to the right by five bits (G_5) in a fourth array of adders 94 to the output of the third array of adders 88. Finally, the most significant bit of F shifted all the way to the least significant position (six bits to the right) is selectively added in a fifth array of adders 96 to the output of the fourth array of adders 94 under the control of line 22, which determines if G_6 or zero is added in at multiplexer array 98 (an array of one multiplexer in this case). The output of the fifth array of adders 98 produces the divided output P.

In the general case, the input value F is shifted successively to produce shifted values G_1 through G_{M-1} where the subscript represents the number of places to the right the binary number F is shifted. The adders produce output values in any combination of sums of the shifted values G_1 through G_{M-1} . Thus, the user can produce any fraction of F as a selected sum of the G values, with the selection being by virtue of selection of a value of G or zero in each multiplexer stage. Therefore, the value of P at the output is given by:

$$P = \sum G_K, \text{ for all selected values of } K.$$

Those skilled in the art will appreciate that no clocking is required to accomplish the selected division. Moreover, since each full adder may be constructed to produce its output after approximately two gate delays, the overall division process proceeds rapidly. The current divider 10 can be

expanded or contracted to produce division of an arbitrary N bit binary number by any power of two from 2 through 2^{N-1} . These divided values can be selectively added together to produce output values ranging from $F/2^{N-1}$ through $F(2^{N-1} - 1)/2^{N-1}$.

Referring now to **FIG. 2**, a full adder in accordance with one embodiment of the present invention is illustrated as adder 200. Such an adder is suitable for any of the adders shown in **FIG. 1** (e.g. 50, 52, etc.) In adder 200, the A and B inputs are simultaneously applied to an exclusive OR (EXOR) gate 202 and an AND gate 206. The output of exclusive OR gate 202 is applied to an exclusive OR gate 208 and simultaneously to an AND gate 212. The carry input (C_{in}) is applied to a second input of both exclusive OR gate 208 and AND gate 212. The outputs of AND gates 206 and 212 are applied to the two inputs of an OR gate 216. The output of exclusive OR gate 208 provides the sum bit S while the output of OR gate 216 provides the carry out bit C_{out} .

Thus, the present invention provides a method and apparatus for selectively dividing an N bit binary integer F by any selected value ranging from 2 through 2^{N-1} ; these values can be selectively added together to achieve any output value ranging from $F/2^{N-1}$ through $F(2^{N-1} - 1)/2^{N-1}$. The divider operation is selectively controlled by controlling the selected input of a plurality of multiplexer arrays with each multiplexer array selectively switching either zeros or a right shifted version of the number to be divided F. In the general case, an array of multiplexers is provided to selectively add either zero or a shifted version of F (G) for each bit of F except the least significant bit which produces a remainder which may be used separately or discarded, depending upon the application. Thus, for an N bit binary number, N-1

arrays of multiplexers such as 26 and 66 are used, with each successive multiplexer array selectively providing the input F shifted by an additional bit until the most significant bit is in the least significant position. Also, in the general case, N-2 arrays of 1 bit adders, such as 44, are used to accomplish the required addition.

The multiplexers may be implemented as simple CMOS (or other process) transmission gates or switches, but this is not intended to be limiting since any suitable switch design can be used whether using bipolar, MOS or other technologies. Those having ordinary skill in the art will also understand that the number of arrays of full adders can be reduced by using appropriate combinations of three input adders or the like without departing from the present invention.

While the invention has been described in conjunction with specific embodiments, it is evident that many alternatives, modifications, permutations and variations will become apparent to those of ordinary skill in the art in light of the foregoing description. Accordingly, it is intended that the present invention embrace all such alternatives, modifications and variations as fall within the scope of the appended claims.

What is claimed is:

1. A programmable divider for dividing an N bit binary number F by a programmed value equal to $(2^{N-1}) / K$, where K is a selected integer between 1 and $(2^{N-1} - 1)$ to produce a divided binary number P, comprising:

N - 1 shift circuits for dividing the N bit binary number F by shifting the N bit binary number F to the right by 1 through (N-1) bits respectively, wherein the shift circuits produce N-1 divided values G_1 through G_{N-1} having values of $F/2$ through $F / (2^{N-1})$ respectively; and

a plurality of adders that add any selected combination of the divided values G_1 through G_{N-1} together to produce the divided binary number P.

2. The programmable divider of claim 1, wherein each of the plurality of adders selectively adds either a divided value G_1 through G_{N-1} or zero.

3. The programmable divider of claim 2, further comprising a plurality of switches that selectively couples either a divided value G_1 through G_{N-1} or zero to an input of each adder.

4. The programmable divider of claim 3, wherein each of the switches comprises a two input multiplexer.



5. The programmable adder of claim 3, wherein the plurality of switches is arranged in $N-1$ arrays to produce zero or one of G_1 through G_{N-1} as an output thereof.
6. The programmable divider of claim 1, wherein each of the plurality of adders comprises a plurality of one bit full adders with carry.
7. The programmable divider of claim 1, wherein the plurality of adders comprise $N-2$ arrays of N one bit adder circuits.
8. A method of dividing an N bit binary number F by a programmed value equal to $(2^{N-1}) / K$, where K is a selected integer between 1 and $(2^{N-1} - 1)$, to produce a divided binary number P , the method comprising:
- shifting the N bit binary number F to the right by 1 through $(N-1)$ bits respectively to produce $N-1$ divided values G_1 through G_{N-1} having values of $F/2$ through $F / (2^{N-1})$ respectively; and
- adding any selected combination of the divided values G_1 through G_{N-1} together to produce the divided binary number P .



9. The method of claim 8, wherein adding comprises summing the selected combination of values G_1 through G_{N-1} with zeros for each unselected value G_1 through G_{N-1} .

10. The method of claim 8, wherein the adding is carried out in $N-2$ arrays of 1 bit full adders.

11. The method of claim 8, wherein the adding is carried out in a plurality of adders that selectively adds either a divided value G_1 through G_{N-1} or zero.

12. The method of claim 11, wherein a plurality of switches selectively couples either a divided value G_1 through G_{N-1} or zero to an input of each adder.

13. The method of claim 12, wherein each of the switches comprises a two input multiplexer.

14. The method of claim 12, further comprising arranging the plurality of switches in $N-1$ arrays to produce zero or one of G_1 through G_{N-1} as an output thereof.



15. The method of claim 11, wherein each of the plurality of adders comprises a plurality of one bit full adders with carry.

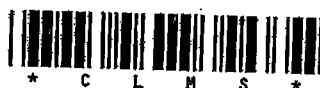
16. The method of claim 11, wherein the plurality of adders comprise N-2 arrays of N one bit adder circuits.

17. A programmable divider for dividing an N bit binary number F by a programmed value equal to $(2^{N-1}) / K$, where K is a selected integer between 1 and $(2^{N-1} - 1)$ to produce a divided binary number P, comprising:

N -1 shift circuits for dividing the N bit binary number F by shifting the N bit binary number F to the right by 1 through (N-1) bits respectively, wherein the shift circuits produce N-1 divided values G_1 through G_{N-1} having values of $F/2$ through $F / (2^{N-1})$ respectively;

a plurality of one bit full adders with carry arranged in N-2 arrays of N one bit full adders with carry, that add any selected combination of the divided values G_1 through G_{N-1} or zero together to produce the divided binary number P; and

a plurality of switches, each comprising a two input multiplexer, arranged in N-1 arrays that selectively couple either a divided value G_1 through G_{N-1} or zero to an input of each adder.



18. The programmable divider of claim 17, wherein the adders and switches are implemented using CMOS circuits.
19. A programmable divider as described herein with reference to Figures 1 and 2.
20. A method of dividing as described herein with reference to Figures 1 and 2.

