SYNCHRONOUS COUNTERS IMPLEMENTED IN THE PLD DEVICES

Jaromír KOLOUCH
Dept. of Radioelectronics
Technical University of Brno
Purkyňova 118, 612 00 Brno
Czech Republic
E-mail: kolouch@urel.fee.vutbr.cz

Abstract

The implementability of synchronous counters in the Programmable Logic Devices (PLD) is discussed in this paper. The most commonly used counters are analysed from this point of view. The expressions for their individual bits are given and the number of product terms is derived to allow to estimate the size of the particular counter which can be implemented in the chosen PLD.

Keywords

programmable logic devices, synchronous counters, product terms, implementability

1. Introduction

The programmable logic devices are very suitable for implementation of various synchronous sequential systems. Most often are used counters with various codes. It is useful to know their requirements for the product terms to be able to estimate the size of the counter which can be implemented in a particular device or to choose the suitable device for the desired counter.

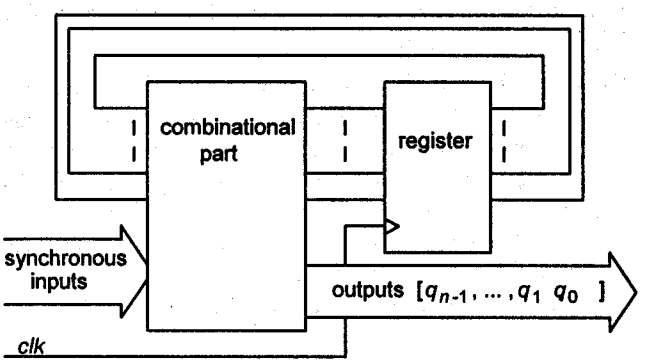

Fig. 1. General connection of the synchronous counter

The general connection of the counter is given in the Fig. 1. In the PLD devices, the combinational part is placed in the programmable AND array of the device. The register may be of the D or T type. The programmable inverters which are found in most PLDs may be included into the combinational part. The behaviour of the counter with the D-type register is described by the function of the next state

$$q' = nxt \tag{1}$$

where *nxt* is the expression for the variable q after the clock pulse. The counter with the T-type register can be described by the formula

$$q.\mathrm{T} = tgl \tag{2}$$

with *tgl* meaning the condition of the toggle of the bit q in the next counter step. Both those expressions can be given in the default (non-inverted) form according to (1) or (2), or in the inverted form which will be marked by the bar upon the symbol. In all cases, the left sides of those relations represent the signals at the output of the AND array of the PLD where the counter is implemented.

If the toggle condition *tgl* is known, the next state can be expressed as

$$q' = q \oplus tgl, \tag{3}$$

with the next state known the toggle condition can be written by the relation

$$q.\mathrm{T} = q \oplus nxt. \tag{4}$$

For the most often used counters, we give here the relations for their bits. Where for the lowest or highest bits are their expressions obvious and are not included in the formulas, we do not write them to save space.

2. Binary Counters

In the case of the *binary up or down full cycle counter* based on the T-type register, the condition tgl_k for the bit b_k (for $k > 0$) can be expressed for the up counter

$$b_k.\mathrm{T} = tgl_k = b_{k-1} \cdot \ldots \cdot b_0, \tag{5}$$

and for the same down counter is

$$b_k.\mathrm{T} = tgl_k = \overline{b_{k-1}} \cdot \ldots \cdot \overline{b_0}. \tag{6}$$

After substitution to (3) we obtain expressions for the D-type register:

$$b_k' = b_k \oplus (b_{k-1} \cdot \ldots \cdot b_0) \tag{7}$$

for the up counter, and

$$b_k' = b_k \oplus (\overline{b_{k-1}} \cdot \ldots \cdot \overline{b_0}) \tag{8}$$

for the down counter. The conversion to the sum-of-product (SOP) form gives expressions with $k+1$ terms.

These expressions are well known and we do not write them here to save space. Therefore, for the n-bit up or down binary counter, n terms are required in the expression for the most significant bit if D-type register is used, and for each lower bit, this number is decreased by one.

For the *binary up/down full cycle counter* based on the T-type register, the condition tgl_k can be expressed (again for $k > 0$)

$$b_k.\mathrm{T} = tgl_k = b_{k-1} \cdot \ldots \cdot b_0 \cdot up + \overline{b_{k-1}} \cdot \ldots \cdot \overline{b_0} \cdot \overline{up}. \quad (9)$$

After substitution to (3) and simplification we find the expression for the bit b_k for the D-type register

$$b_k' = \overline{b_k} \cdot (b_{k-1} \cdot \ldots \cdot b_0 \cdot up + \overline{b_{k-1}} \cdot \ldots \cdot \overline{b_0} \cdot \overline{up}) + \\ + b_k \cdot (b_{k-1} \cdot b_{k-2} + \ldots + b_0 \cdot up + up \cdot b_{k-1}). \quad (10)$$

Therefore, the n-bit full cycle up/down binary counter requires $n + 2$ terms for the most significant bit. For each lower bit, this number is decreased by one as in the case of the one-directional counter. If the popular GAL 16V8 device is used, 6-bit up/down counter can be directly implemented. The two remaining macrocells can be used to add one further counter bit in two-pass configuration.

From the expressions for the T-type register it follows that only one term is necessary for the one-directional binary counter and two terms are sufficient for the up/down counter of any width.

The expressions for the inverted form require the same number of terms as for the default form in the case of both register types.

The *binary up counters with shortened cycle* contain unused (illegal) states and can therefore be built by more ways. The various modifications of these counters with the same cycle length can differ in the states which are deleted from the full cycle and/or in the behaviour in those unused states. From the point of view of the implementability of those counters in the PLD devices, it is important to find the modifications that require the smallest possible number of terms, or at least do not require far more terms than the full cycle counters.

The last condition is fulfilled by the up counter which can be obtained from the full cycle counter by deleting some number of its highest states. In those unused states, this counter counts up to the highest state of the full cycle counter and then returns to zero. Therefore, it differs from the full cycle counter only in the behaviour in its last state (let this state be numbered l) where it goes to zero whereas the full cycle counter increments. For the n-bit counter, the values of l in the range from $2^{n-1} + 1$ to $2^n - 2$ are meaningful but the further considerations are valid for any l in the range from 1 to $2^n - 1$, it is for any such possible n-bit shortened or full cycle binary up counter.

For the bit b_k ($k > 0$) of such a counter with D-type register, the expression is given by the expression for the same bit of full cycle counter which is multiplied by the maxterm d_l:

$$b_k' = (b_k \oplus (b_{k-1} \cdot \ldots \cdot b_0)) \cdot d_l. \quad (11)$$

If this expression is written in the inverted form, it can be easily recognized that it requires for any bit by one term more than the expression for the full cycle counter - according to the de Morgan's rule it is the sum of that expression and of the minterm k_l which represents the inversion of the maxterm d_l. For some values of l, it is still possible to reduce the number of terms but this reduction is only small. As the number of required terms for the full cycle counter equals to the order of the bit in question, only the most significant bit could cause impossibility to implement counters with shortened cycle in the devices where full cycle counter of the same bit number fits. Fortunately, it can be proved that this bit when expressed in the default form requires just n terms for any n-bit counter. Therefore, we can conclude that any n-bit binary up counter of this type with any possible number of states can be implemented in a PLD device with n terms in macrocell if this device contains the programmable inverter.

If the lower bits are expressed in the default form, it is not simple to give a general rule for the number of terms. It can be verified by the computer at least for 8-bit binary counter with shortened cycle that any bit of this counter expressed in this form does not require more terms than the most significant bit. This statement seems to be valid for counters with other bit width as well.

Similarly as for the full-cycle counters, the use of the T-type register in the case of shortened cycle binary counters can dramatically decrease the required number of terms. It follows from the expression (11) for the D-type register and from the formula (4) that the expression for the shortened cycle binary counter bits with T-type register with no inverter included before the register input is

$$b_k.\mathrm{T} = b_{k-1} \cdot \ldots \cdot b_0 \cdot (b_k + d_l) + b_k \cdot k_l. \quad (12)$$

The minimization shows this expression contains one or two terms except to the bit whose order is equal to the order of the lowest zero bit in the number l written in binary notation. If this order is m, the number of terms required for the bit b_l equals $n - m$. If inverter is included before this register input, the required number of terms is $m + 1$. For small values of the number m, the inverted expression for the bit b_l is more advantageous.

There are *shortened cycle binary down counters* which count from the all-ones state to some terminating non-zero state. They can be obtained from the discussed up counters simply by the inversion of their outputs. The expression for the bit b_k can be written if the D-type register is used:

$$b_k' = (b_k \oplus (\overline{b_{k-1}} \cdot \ldots \cdot \overline{b_0})) + k_l. \quad (13)$$

As to the number of terms, the same rules are valid if the default and inverted form of expressions are exchanged. This is valid for the use of the T-type register as well.

The T-type register can be found e. g. in the EP 610 device. Here, 16-bit full cycle or nearly arbitrarily shortened cycle counter can be designed. As the T-type register can be emulated by the D-type register with hardware EX-OR gate, devices containing this gate, e. g. the ispLSI series devices of Lattice can be used in the same way.

3. BCD-Code Counters

In those counters, unused states occur as well. Similarly as for the shortened cycle binary counters, there are several modifications with different behaviour in the unused states and different requirements for the number of terms. Some of them will be described here.

Let us consider at first the subgroup of those counters which is characterized by the fact that individual decades, i. e. groups of four bits expressing individual decimal digits, have identical behaviour in the unused states. If we give the behaviour of the lowest decade e. g. by the corresponding state diagram, then the same state diagram is valid for any of the higher decades. Every higher decade changes its state if and only if all lower decades are in the state 9. Based on this observation, the expressions for the individual bits can be found.

If as low number of terms required as possible is desired, following expressions for the bits of the k-th order decade can be used:

$$\left.\begin{aligned} d_{k0}.\mathrm{T} &= P_k, \\ d_{k1}.\mathrm{T} &= d_{k0}\cdot\overline{d_{k3}}\cdot P_k, \\ d_{k2}.\mathrm{T} &= d_{k0}\cdot d_{k1}\cdot P_k, \\ d_{k3}.\mathrm{T} &= (d_{k0}\cdot d_{k3} + d_{k0}\cdot d_{k1}\cdot d_{k2})\cdot P_k. \end{aligned}\right\} \quad (14)$$

where

$$P_k = \prod_{i=0}^{k-1} d_{i0}\cdot d_{i3} \quad \text{for } k>0 \qquad (15)$$

with $P_0 = \mathbf{I}$. The state diagram of one decade is drawn on the Fig. 2. The k-th decade transitions if the condition P_k is true. It is apparent the number of terms is 1 for the lowest three bits of any decade and 2 for the highest bit. The n-bit counter returns to the working cycle after 2.10^{n-1} steps in the most unfavourable case, i. e. from the states 14, 12 or 10 of the highest decade and cleared lower decades.

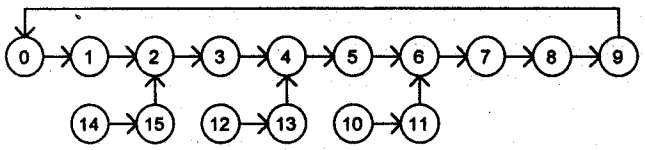

Fig. 2. State diagram of one decade according to (14)

If it is desired to decrease the number of steps required for the return to the working cycle, the expression for the most significant bit of a decade can be changed to

$$d_{k3}.\mathrm{T} = (d_{k0}\cdot d_{k3} + d_{k0}\cdot d_{k1}\cdot d_{k2} + d_{k1}\cdot d_{k3} + d_{k2}\cdot d_{k3})\cdot P_k. \qquad (16)$$

It can be found the counter then needs in the worst case only 10^{n-1} steps to return to the working cycle. Further modifications are possible if it is desired to balance the number of terms among the bits of the decade with the same number of steps for the return to the working cycle.

It is characteristic for these counters and advantageous for their implementability that the bits for all decades are expressed by the same number of terms, e. g. for the counter according to (14) by one or two terms.

Other modification with practical meaning is the decade counter which returns to the working cycle immediately from any unused state. It is based on the counter described by the relations (14) where a correction expression to the bit d_{k3} is added which clears this bit if the value of corresponding decade is out of the correct range:

$$d_{k3}.\mathrm{T} = (d_{k0}\cdot d_{k3} + d_{k0}\cdot d_{k1}\cdot d_{k2})\cdot P_k + d_{k3}\cdot(d_{k2} + d_{k1}) \qquad (17)$$

Apparently, this correction adds two more terms. Therefore, four terms are necessary for this bit.

The state diagram of one decade can be now drawn according to Fig. 3. The transitions in the working cycle occur only if the condition P_k is true, but the return from any illegal state takes place immediately in the next step. If the condition P_k for the decade which is in an illegal state is false, only its highest bit is cleared as is indicated by the dashed line in upward direction. Otherwise, the lower bits are changed and the return is indicated by the full line.

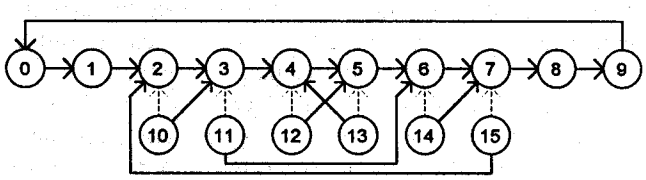

Fig. 3. State diagram of one decade for the counter returning to the working cycle from any illegal state. Dashed line is for P_k = O

In the EP 610 device, 16-bit decade counter of this type can be easily implemented.

4. Gray-Code Counters

These counters are often used in applications where the most known Gray-code property, i. e. change of only one bit in each step, is advantageous.

For the classical design approach to the n-bit counter, following expressions for the up/down counter bits in devices with the D-type register can be derived:

$$\left.\begin{aligned} g_0' &= g_1 \oplus \ldots \oplus g_{n-1} \oplus up, \\ g_1' &= g_1 \oplus [g_0\cdot(g_1 \oplus \ldots \oplus g_{n-1} \oplus up)], \\ &\cdots\cdots\cdots \\ g_k' &= g_k \oplus [\overline{g_0}\cdot\ldots\cdot\overline{g_{k-2}}\cdot g_{k-1}\cdot(g_k \oplus \ldots \oplus g_{n-1} \oplus up)] \quad \text{for } k = 2, \ldots, n-2, \\ &\cdots\cdots\cdots \\ g_{n-1}' &= g_{n-1} \oplus [\overline{g_0}\cdot\ldots\cdot\overline{g_{n-3}}\cdot(g_{n-2} \oplus g_{n-1} \oplus \overline{up})]. \end{aligned}\right\} \quad (18)$$

The number of terms for the bit g_k is

$$t_k = 2^{n-k-1} + k. \qquad (19)$$

For the up or down counter, this number is n - 1 for g_{n-1}, and for $k < n$ - 1 is

$$t_k = 2^{n-k-2} + k. \qquad (20)$$

It is not difficult to prove the number of terms is the same for the default as well as for the inverted form.

For the devices with the T-type register, the expressions are

$$\left.\begin{aligned} &g_0 .\mathrm{T} = g_0 \oplus \ldots \oplus g_{n-1} \oplus up, \\ &g_1 .\mathrm{T} = g_0 \cdot (g_1 \oplus \ldots \oplus g_{n-1} \oplus up), \\ &\quad \ldots\ldots\ldots\ldots\ldots \\ &g_k .\mathrm{T} = g_0 \cdot \ldots \cdot g_{k-2} \cdot g_{k-1} \cdot (g_k \oplus \ldots \oplus g_{n-1} \oplus up) \\ &\qquad \text{for } k = 2, \ldots, n-2, \\ &\quad \ldots\ldots\ldots\ldots\ldots \\ &g_{n-1} .\mathrm{T} = \overline{g_0 \cdot \ldots \cdot g_{n-3} \cdot (g_{n-2} \oplus g_{n-1} \oplus up)}. \end{aligned}\right\} \quad (21)$$

For the bit g_{n-1}, four terms are necessary, and the number of terms for $k < n - 1$ is

$$t_k = 2^{n-k}. \quad (22)$$

The up or down counter requires half the number of terms for every bit.

If the expressions (21) are inverted, the number of terms for the inverted bit g_{n-1} is $n + 2$ and for the inverted bit g_k ($k < n - 1$) is

$$t_k = 2^{n-k} + k. \quad (23)$$

To summarize, the growth of the number of terms is exponential with the number of the counter bits in all cases. Therefore, this design approach is applicable for the Gray-code counters with a small number of bits only.

4.1 Gray-code counters with auxiliary bit

The number of terms required for the bits of the Gray-code counter can be considerably decreased by the use of the auxiliary bit approach. The auxiliary bit completes the code word represented by the counter contents to the even or odd parity. According to this, symbol *pe* or *po* will be used for it.

The expressions for the full cycle up/down counter bits with the odd-parity auxiliary bit and the T-type register can be:

$$\left.\begin{aligned} &g_0 .\mathrm{T} = po \oplus \overline{up}, \\ &g_1 .\mathrm{T} = g_0 \cdot \underline{(po \oplus up)}, \\ &g_2 .\mathrm{T} = g_1 \cdot g_0 \cdot (po \oplus up), \\ &\quad \ldots\ldots\ldots\ldots\ldots \\ &g_k .\mathrm{T} = g_{k-1} \cdot g_{k-2} \cdot \ldots \cdot \overline{g_0} \cdot (po \oplus up) \\ &\qquad \text{for } k = 3, \ldots, n-2, \\ &\quad \ldots\ldots\ldots\ldots\ldots \\ &g_{n-1} .\mathrm{T} = g_{n-3} \cdot \ldots \cdot g_0 \cdot (po \oplus up). \end{aligned}\right\} \quad (24)$$

If the even-parity auxiliary bit is used, the relations (24) are still valid, only the parity bit *pe* is substituted for the *po* bit and this bit is inverted.

The number of terms required for this counter is two in the expression of each up/down counter bit. For the one-directional up or down counter, only one term per bit is necessary.

If the D-type register is used, the corresponding relations can be found from (24) according to (3):

$$\left.\begin{aligned} &g_0' = g_0 \oplus po \oplus \overline{up}, \\ &g_1' = g_1 \oplus (g_0 \cdot \underline{(po \oplus up)}), \\ &g_2' = g_2 \oplus (g_1 \cdot g_0 \cdot (po \oplus up)), \\ &\quad \ldots\ldots\ldots\ldots\ldots \\ &g_k' = g_k \oplus (g_{k-1} \cdot g_{k-2} \cdot \ldots \cdot \overline{g_0} \cdot (po \oplus up)) \\ &\qquad \text{for } k = 3, \ldots, n-2, \\ &\quad \ldots\ldots\ldots\ldots\ldots \\ &g_{n-1}' = g_{n-1} \oplus (g_{n-3} \cdot \ldots \cdot \overline{g_0} \cdot (po \oplus up)). \end{aligned}\right\} \quad (25)$$

Here, the number of terms for the bit g_k is $k + 4$ for $k = 0$ up to $n - 2$ and it is $n + 2$ for the most significant bit g_{n-1}. Compared to the up/down binary counter, the number of terms is the same for the most significant bit and only by one higher for other bits except the least significant bit. This is great advantage over the classical design approach.

The auxiliary bit can be created either by the combinational logic from other counter bits or it can be stored in a separate flip-flop (macrocell).

In the first case, the auxiliary bit *po* can be obtained as inverted EX-OR function of all the counter bits:

$$po = \overline{g_{n-1} \oplus \ldots \oplus g_0}. \quad (26)$$

In this case, the auxiliary bit *po* represents the odd parity. If the right-hand side expression is not inverted, the even parity bit *pe* is obtained instead. The relation (26) can be added to the relations (24) and (25). The number of terms required for (26) grows exponentially with the number of counter bits. The number of terms for other counter bits is nevertheless more favourable, only the auxiliary bit exhibits this exponential dependence. If T-type register and two-pass configuration for the auxiliary bit are used, up to 12-bit up/down Gray-code counter can be implemented in the EP 610 device. As the auxiliary bit is used in the expressions for the other bits, total number of passes is three which considerably decreases the maximum counter frequency.

The second possibility is based on the fundamental property of the Gray-code counter, only one-bit change in every counter step. This means the parity of the counter is changed in every step. If an additional bit which changes its value in every step is added to the counter, the total parity of the counter with this bit remains without change. This is just the property of the auxiliary bit defined above. To complete the relations (24) or (25), one of following relations according to the type of register used should be added to them:

$$po .\mathrm{T} = \mathrm{I} \quad \text{or} \quad po' = \overline{po}. \quad (27)$$

Here, only one term is used for the auxiliary bit. In this way, Gray-code counter with T-type register can be implemented without any limitations to the number of its bits due to the number of terms. With the D-type register, the limitation is approximately the same as for the binary counters, only the auxiliary bit must be added, which requires one macrocell in the PLD devices.

Table 1. Survey of the implementability of counters in the most widely used PLD devices

counter type	PLD type / max. number of bits possible for the counter					
	GAL 16V8	GAL 20V8	GAL 18V10	GAL 22V10	GAL 20XV10	EP 610
one-directional binary	8 bits	8 bits	10 bits	10 bits	10 bits	16 bits
bi-directional binary	6 (7) bits	6 (7) bits	8 (9) bits	10 bits	10 bits	16 bits
one-directional binary with shortened cycle	8 bits	8 bits	9 bits	10 bits	not suitable	16 bits (see text)
one-directional Gray without unused states	5 bits	5 bits	5 (6) bits	6 (7) bits	not suitable	5 (12) bits
bi-directional Gray with auxiliary bit	7 bits	7 bits	9 bits	9 bits	9 bits	15 bits

In table 1, survey of the possibilities to implement various counter types in the common PLD devices is given. Where two-pass configuration allows to implement more counter bits, corresponding number of bits is given in the parentheses. In the EP 610 device, most (perhaps all) of the 16-bit one-directional shortened cycle binary counters can be implemented. Their configuration must be chosen individually. A lot of them were tried and found to be implementable in this device. Five-bit Gray counter can be designed in any device with 8 terms per macrocell, two-pass 12-bit one with auxiliary bit (one-directional or reversible) can be implemented in EP 610 as mentioned above.

If the auxiliary bit in the Gray counter is stored in a flip-flop of a separate macrocell, the number of states of this counter is twice that of the used (legal) states. It can be found the illegal states form one illegal cycle in which the counter counts in Gray code in the opposite direction than in the legal cycle. If the counter falls anyhow into some of those states, it continues counting in this cycle till it is forced from outside to the legal one.

It is often desired that the counter returns automatically to the legal cycle when it falls by chance into the illegal one. This can be ensured by a correction of the expressions for some counter bits. The principle may consist in disabling the change of some bit, e. g. the auxiliary bit, if the unused state is detected. For example, if the odd parity is used, the expression for the auxiliary bit can be modified as follows:

$$po' = \overline{po} \oplus (\overline{g_{n-1}} \cdot \ldots \cdot \overline{g_0} \cdot \overline{po}). \qquad (28)$$

If the counter comes to all-zero state with the *po* bit zero, the parity is even. Therefore, this state is illegal. The expression (28) for the auxiliary bit ensures the zero next value of this bit. The parity is now correct - the counter is in the legal cycle again. The illegal state detection can be extended to more states to reduce the number of illegal states the counter passes before it returns to the legal cycle.

5. Conclusions

The most common counter types were analysed from the point of view of their implementability in the PLD devices. In case of the counters with unused states, modifications of those counters suitable for the implementation were found.

The results are stated here for the simple PLD devices, they are nevertheless applicable for the complex PLDs as well. They enable the designer to estimate the amount of the device resources required for the implementation of those counters.

6. References

[1] PELLERIN,D.-HOLLEY,M.: Digital Design Using ABEL. Prentice Hall, Englewood Cliffs, New Jersey 07632, 1994

[2] KOLOUCH,J.: On the implementability of synchronous counters in common PLD devices. In: *„Radioelektronika 98", Conference Proceedings*, Vol. 2, pp. 77-80. TU in Brno, Brno 1998

[3] KOLOUCH,J.: Gray-code counters in the PLD devices. In: *„EDS 98", Conference Proceedings*, Vol. 2, pp. 398-401. TU in Brno, Brno 1998

About the author...

Jaromír KOLOUCH was born in Brno in 1945. He received the Ing (MSc) degree from the Military Academy in Brno in 1968 and CSc (PhD) degree from the Technical University in Brno in 1984. He is interested in the digital technique, especially in the programmable logic devices.