A SYSTEMATIC APPROACH TO THE DESIGN OF STRUCTURES FOR ADDITION AND SUBTRACTION
_ CASE OF RADIX $r = m^k$ -

Diem Dinh Nguyen

University of Illinois at Urbana-Champaign
Urbana,IL, 6I80I

ABSTRACT

The results of Robertson concerning a systematic approach to the design of Adder/Subtracter structures of radix $r = 2^k$, $k \geq I$ are generalised to cover all structures of radix $r = m^k$, $k \geq I$ and $m \geq 2$ The use of Quasibinary representations help reduce the number of types of fundamental structures required . In addition to the types encountered in the earlier case , only one new type of fundamental structures called Radix-m Carry Generator is needed . Examples in the particular case of Decimal Adder/Subtracter structures are used to illustrate the results .

I.Introduction

A design theory for a class structures for Addition and Subtraction is presented in a recent paper [I] . An Adder/Subtracter structure is one that transforms one or more input operands into one or more output operands where each input and output operand is a digit set , i.e. a consecutive sequence of integers including zero and the transformation is subjected to some precise rules . In essence, the above paper demonstrates that a complex Adder/Subtracter structure can be realised as combinations of simpler fundamental structures with known or more easily determined implentations . The input and output operands of each type of fundamental structures are either two-valued or three-valued digit sets .

A distinguishing feature for this design theory is the partition of given digit sets with more than three values into binary weighted two-valued and three-valued digit sets . Such a partition is referrred to in this paper as a Quasibinary representation , in contrast to the conventional Binary representations which consist only of binary weighted two-valued digit sets . Because a three-valued digit set can be replaced by two two-valued digit sets , Quasibinary representations subsume Binary representations as special cases . Furthermore , because the three-valued digit set is a redundant digit set with respect to the basic radix $r = 2$ of the fundamental structures mentioned above , the use of Quasibinary representations incorporates in a natural way this type of redundancy which is known to lead to improvements in the design of arithmetic structures [2-8].

The design process consists of two steps . The first step involves the decomposition of the given structure into fundamental structures and the second step the filling in of implementation details of the fundamental structures being used . There also exist relations that provide partial measures of the hardware costs and some guidelines in the decomposition step of the design process.

The main restriction on the results of [I] is that the Adder/Subtracter structures to be designed must have radix r of the form $r = 2^k$, $k \geq I$. Since some Adder/Subtracter structures of intrest do not fall into the above class (one eminent case being the class of Decimal Adder/Subtracter structures) , this paper generalises those results to deal with Adder/Subtracter structures of radix $r = m^k$, $k \geq I$ and $m \geq 2$.

As will be seen , the Quasibinary representation will again assume an important role . Its use leads to a considerable reduction in the number of types of fundamental structures required in the decomposition step . The set of fundamental structures for the case of radix $r = m^k$, $k \geq I$, $m > 2$ contains those same fundamental structures for the case of radix $r = 2^k$, $k \geq I$, plus a new type called Radix-m Carry Generator . There exist other differences between the two cases ; some of these differences appear in examples concerning designs of Decimal Adder/Subtracter structures of radix $r = IO^k$, $k \geq I$. This class of structures is considered not only to illustrate the theory but also because of its possible application value .

Due to space limitations , the proofs of all results together with many clarifying illustrations will be omitted . The

CH1630-3/81/0000/0042$00.75 © 1981 IEEE

reader is referred to [9] for details . The paper is organised as follows. Section 2 reviews the basic concepts and notations Section 3 contains the main results , Section 4 presents some illustrative examples and Section 5 provides some concluding remarks .

2. Digit sets and Adder/ Subtracter structures

A $(\delta+1)$-valued digit set is a sequence of $\delta+1$ consecutive integers including zero $\{-w,-w+1,\ldots,-w+\delta\}$ where $0 \leq w \leq \delta$. δ is called the diminished cardinality and w the offset of the digit set . Let r be the radix involved, a digit set is redundant if $\delta > r-1$, nonredundant if $\delta = r-1$ .Since r can be as low as 2 , the lower limit of δ is 1. There is no upper limit for δ but we use the arbitrary upper limit $2(r-1)$ which is sufficiently large for most practical purposes . The digit set is said to be normalised if $w=0$, symmetric if δ is even and $w = \frac{\delta}{2}$.

Digit sets will be denoted by letters of the alphabet with a superscript indicating the offset . For example $a^0 \equiv \{0,1\}$, $a^1 \equiv \{\bar{1},0\}$, $b^0 \equiv \{0,1,2\}$, $b^1 \equiv \{\bar{1},0,1\}$, $b^2 \equiv \{\bar{2},\bar{1},0\}$, ... , $i^Q \equiv \{0123456789\}$ ... It is important that the three-valued digit set b^w (also any digit set with larger number of values) be recognised as a separate entity , independent of its binary representation . It is also useful to refer to the class of digit sets of a given diminished cardinality without specifying the offset by omitting the superscript in the digit set symbol .

A set of integers A is a Setsum of the sets of integers B and C if and only if for every $b \in B$ and $c \in C$ there exists an $a \in A$ such that $a=b+c$. Using the symbol '$\dot{+}$' for Set Addition operation , we can write either $A \equiv B \dot{+} C$ or $A \equiv C \dot{+} B$ due to its commutative property .

A Quasibinary representation of a given digit set is a partition into binary weighted two-valued digit sets (a) and three-valued digit sets (b) such that the Setsum of these binary weighted components equals the original digit set . A simple algorithm to find such representations is given in [1]:

Algoritm: Given a digit set δ^w with $\delta > 2$

0) (Ignore the offset w till Step 2) INITIALISE $i \leftarrow 0$; $\delta_i \leftarrow \delta$.

1) WHILE $\delta_i > 0$ DO

BEGIN IF δ_i is ODD , THEN include $2^i a$ as a component; $\delta_{i+1} = \frac{1}{2}(\delta_i - 1)$. ELSE include $2^i b$ as a component, $\delta_{i+1} = \frac{1}{2}(\delta_i - 2)$. $i \leftarrow i+1$.

END .

2) IF $w=0$, THEN assign offset 0 to all components ; ELSE ($w>0$) assign offsets to individual components in the representation found above so that their weighted sum is equal to w (Note: There may be more than one way to do this).

3) END.

For example , without taking offset into account , the digit set n (with $\delta = 14$) has the Quasibinary representation $4b \dot{+} 2b \dot{+} b$ .Thus the normalised digit set n° has the unique Quasibinary representation $4b^0 \dot{+} 2b^0 \dot{+} b^0$. In contrast , the digit set n^4 can have either $4b^1 \dot{+} 2b^0 \dot{+} b^0$ or $4b^0 \dot{+} 2b^2 \dot{+} b^0$ or $4b^0 \dot{+} 2b^1 \dot{+} b^1$ as a Quasibinary representation .

A representation is said to overrepresent a digit set if it can represent values other than those belonging to the digit set in question . A close examination of the definition and formation of Quasibinary representations show that the problem of overrepresentation will never occur . In comparison , such a problem may occur when one uses Two's Complement binary representations (and other well-known Binary representations) . For example , the symmetric digit set $\{\bar{8},\bar{7},\ldots,0,\ldots,7,8\}$ has a Quasibinary representation $8a^1 \dot{+} 4a^0 \dot{+} 2a^0 \dot{+} b^0$, but for Two's Complement binary representation , one is forced to use $\bar{1}6a^1 \dot{+} 8a^0 \dot{+} 4a^0 \dot{+} 2a^0 \dot{+} a^0$ which can represent also the values $(\bar{1}6,\bar{1}5,\ldots,\bar{9})$ and $(9,10,\ldots,15)$ not in the desired digit set . If the digit set is an input operand , overrepresentation does not cause any problem in logical design . In fact , it helps simplify the result because the so-called overrepresentations are simply DON'TCARES . If however the digit set is an output operand , overrepresentation leads to additional constraint to the logical design problem so that the combinations corresponding to the DON'TCARES will never appear at the the output . In practice , sometimes overrepresentation of a digit set may be used deliberately to gain some saving in bit storage ; the effort in modifying the design and/or the extra hardware cost to achieve this may be justified in those cases [1] .

Because an Adder becomes a Subtracter by a change of sign or by changing the value of a single input and because some structures in the following discussion possess a mixture of properties of Adder and Subtracter and are difficult to characterise , we shall use the terminology of Addition to refer collectively to all struc-

tures . An Adder structure transforms one or more digit sets (inputs) into one or more digit sets (outputs) such that the Setsum of the outputs contains that of the inputs . To help make the decomposition step in the design process more uniform , we adopt the convention that if the above Set Containment is proper a Mythical Input digit set , to the extent necessary to make the Input Setsum equal to the Output Setsum ,will be introduced as shown in Fig. I . Because the Mythical Inputs are just dummy inputs , they can be used to simplify the structure in the final stage of the design process . The introduction of Mythical Inputs therefore does not increase complexity of the final design . Fig. 2 presents a typical example .

Each Adder structure has a negative found by replacing each of the input and output digit sets by its negative . If all input and output digit sets are nonredundant , the logical designs of a structure and its negative are identical . For a structure with redundant digit sets , difficulties are apparently encountered with symmetric digit sets but these can be easily resolved[1].

In the following , an Adder structure will be denoted by a symbol consisting of the Setsum of the inputs on the right side and that of the outputs on the left side of the transformation sign '$\leftarrow$' . For example , the class of Adder structures of Fig. 3.a is designated by $4a + 2a + a \leftarrow 2b + 2a + a$ while one of its variant , shown in Fig. 3.b , is designated by $4a^I + 2a^0 + a^0 \leftarrow 2b^I + 2a^I + a^0$.

3.General results

The first four results of this section show different types of fundamental structures required in the decomposition of Adder structures of radix $r = m^k$, $k \geq I$ and $m \geq 2$. The last result gives a set of relations between the numbers of different types of fundamental structures involved in the decompositions and various parameters associated with the original Adder structure .

Result I : Any Adder structure of radix $r = m^k$, $k \geq I$, $m \geq 2$ can be decomposed into at least one combination of the following types of fundamental structures.

$$ma + [\text{Digit set with } \delta = m-1] \leftarrow [\text{Digit set with } \delta = 2(m-1)] + a$$

$$[\text{Digit set with } \delta = 2(m-1)] \leftarrow [\text{Digit set with } \delta = 2m-3] + a$$

. . . . . .

$$c \leftarrow b + a$$

$$b \leftarrow a + a$$

$$a + a \leftarrow b$$

$$b + a \leftarrow c$$

. . . . . .

$$[\text{Digit set with } \delta = 2m-3] + a \leftarrow [\text{Digit set with } \delta = 2(m-1)]$$

$$[\text{Digit set with } \delta = 2(m-1)] + a \leftarrow ma + [\text{Digit set with } \delta = m-1]$$

Note that the digit sets involved have diminished cardinalities ranging from I up to (2m-I) . The presence of allowed digit sets with $\delta \geq 2$ creates some practical problems . It has been shown that a digit set b ($\delta = 2$) when implemented by two binary variables (two bits) has 9 different formats ; each one represents an NP equivalent group and may lead to a different logical design and hence need be considered individually[1]. A digit set with $\delta > 2$ (such that $\delta \neq 2^k - I$) will have many more formats . For example , a digit set d ($\delta = 4$) when implemented with 3 bits has 30236 formats while a digit set e ($\delta = 5$) has 20580 formats . The task of carrrying out logical designs for all possible versions of a type of fundamental structure corresponding to all formats of allowed digit sets soon becomes enormous and impractical . This in turn makes the selection of one among those formats to be used more difficult . To alleviate this problem , it is decided to replace all allowed digit sets of the given Adder structure by their Quasibinary representations . We then only have to deal with the digit set a with a single format and the digit set b with 9 different formats as mentioned above .As a consequence of the decision to use Quasibinary representations ,we also obtain a reduction in the number of different types of fundamental structures .

Result 2: Under the decision to use Quasibinary representations , the set of fundamental structures of Result I becomes

$$ma + Q_{(m-I)} \leftarrow Q_{(2m-I)}$$

$$2a + a \leftarrow b + a$$

$$b \leftarrow a + a$$

$$a + a \leftarrow b$$

$$b + a \leftarrow 2a + a$$

$$Q_{(2m-I)} \leftarrow ma + Q_{(m-I)}$$

where Q_t the Quasibinary representation of the digit set with the diminished cardinality $\delta = t$

A further reduction follows if we use the so-called FanOut Operators , which in this paper take the general form $n_I a + n_2 a \leftarrow n_3 a$ with $n_I, n_2, n_3 > 0$ and $n_I + n_2 = n_3$.

Result 3 : An equivalent set of fundamental structures to that of Result 2 is

$$ma + Q_{(m-1)} \leftarrow Q_{(2m-1)}$$
$$2a + a \leftarrow b + a$$
$$b \leftarrow a + a$$
$$a + a \leftarrow b$$
$$n_1 + n_2 \leftarrow n_3 \text{ with } n_1, n_2, n_3 > 0 \text{ and } n_1 + n_2 = n_3$$

Thus , in addition to the fundamental structures of the four types (Radix-2 Carry Generator , Generalised Half-Adder, Converter and FanOut Operator) which are encountered in the decomposition of Adder structures of radix $r = 2^k$, $k \geq 1$ we need only a new type , $ma + Q_{(m-1)} \leftarrow Q_{(2m-1)}$, called the Radix-m Carry Generator, for the general case of radix $r = m^k$, $k \geq 1$, $m \geq 2$. As an illustration , for the class of Decimal Adder structures ($m = 10$) ,the set of fundamental structures is

$$10a + 4a + 2b + a \leftarrow 8a + 4b + 2a + a$$
$$2a + a \leftarrow b + a$$
$$b \leftarrow a + a$$
$$a + a \leftarrow b$$
and $$n_1 + n_2 \leftarrow n_3 \text{ with } n_1, n_2, n_3 > 0 \text{ and } n_1 + n_2 = n_3.$$

If in particular m is not a prime integer but of the form $m = 2^q.p$ where $q \geq 1$ and p is a prime integer , we also have

Result 4 : If $m = 2^q.p$ with $q \geq 1$ and p a prime integer , the set of fundamental structures for Adder structures of radix $r = m^k$, $k \geq 1$, $m \geq 2$ is the same as that for Adder structures of radix $r = p^k$, k I with p as defined.

For the illustrative case of Decimal Adder structures ($m = 10 = 2^1.5$), Result 4 means that ,in the set of fundamental structures , the Radix-10 Carry Generator $10a + 4a + 2b + a \leftarrow 8a + 4b + 2a + a$ is replaced by the Radix-5 Carry Generator $5a + 2a + b \leftarrow 4a + 2b + a$.

Let α_{in} (α_{out}) and β_{in} (β_{out}) denote the number of a digit sets and bdigit sets respectively on the input side (output side) of an Adder structure . Let λ_{in} (λ_{out}) denote the "information content " of the input (the output) of the same Adder structure . Following [1] , each a digit set is assumed to have one unit of information while each b digit set two units for convenience .Let $\Delta\alpha \triangleq \alpha_{in} - \alpha_{out}$, $\Delta\beta \triangleq \beta_{in} - \beta_{out}$ and $\Delta\lambda \triangleq \lambda_{in} - \lambda_{out}$. The $\Delta\alpha$, $\Delta\beta$, and $\Delta\lambda$ values for each type of fundamental structures can be easily found For the Radix-m Carry Generator ,until the the specific value of m is known ,we do not know the corresponding $\Delta\alpha$, $\Delta\beta$ and $\Delta\lambda$; we use the generic symbols α_m, β_m and λ_m respectively to refer to these values . Also, define δ_m, ε_m as the number of a digit sets and b digit sets respectively which have weights that are multiples of m on the output side of an Adder structure of radix $r = m^k$, $k \geq 1$, $m \geq 2$ and define $\bar{\delta}_m = \delta_m + 2\varepsilon_m$. Finally let z_m , z , y , x and u denote the number of each Radix-m Carry Generators, Radix-2 Carry generator, Generalised Half-Adders, Converters and FanOut Operators respectively being used in a decomposition . We have :

Result 5 : The following relations hold for any decomposition of an Adder structure of radix $r = m^k$, $k \geq 1$, $m \geq 2$ using the fundamental structures of Result 3

$$z_m \geq \bar{\delta}_m$$
$$z + \lambda_m z_m - u = \Delta\lambda$$
$$y - x + (\alpha_m + \beta_m) z_m - u = \Delta\lambda - \Delta\beta$$

If the only goal is to minimise hardware cost by reducing the numbers of each type of fundamental structures used in a decomposition , the above relations indicate that one should try to use as few Converters and FanOut Operators as possible . Each increase in u leads to an increase in either z or z_m or y while each increase in x causes an increase in y . In practice , another goal in the final design is the total Time Delay due to logic gates , which is usually in conflict with the above goal and forces the designer to select a compromise . Now for a given Adder structure , the parameters $\Delta\beta$, $\Delta\lambda$ and are fixed (note that by definition $\Delta\alpha = \Delta\lambda - 2\Delta\beta$)and the above relations can serve another function as a rough measure of hardware cost or help establish lower and upper bounds on this cost as has been done in [1] .

For the particular case of Decimal Adder structures ($m = 10 = 2^1.5$) , the above relations take the forms $z_5 \geq \bar{\delta}_{10}$, $z - u = \Delta\lambda$, and $y - x - u = \Delta\lambda - \Delta\beta$ since the Radix-5 Carry Generator is being used and $\alpha_5 = \beta_5 = \lambda_5 = 0$.

4. Examples of some Decimal Adder structures Designs

We consider designs of Decimal Digit Slices of Decimal Adders with the Limited-Carry-Propagation property . Each slice is constrained to have the form shown in Fig. 4 , where X,Y,Z and the Mythical Input are the operands and $T_{I(in)}$, $T_{2(in)}$, $T_{I(out)}$, and $T_{2(out)}$ are two-valued Transfer In and Transfer Out signals respectively [1-6] .

The decomposition step can be summerised using an Information Loss Chart [1] which provides traces of important parameters in the decomposition . These parameters are the quantities α , β , λ , the number of a digit sets and b digit sets of the same weights (together with their

offsets) that are available at each stage and the numbers of each type of fundamental structures to be used in that stage . For examples , the symbol 2^1 in the column 8a implies that there exist one $8a^0$ and one $8a^1$ digit sets , the symbol I^1 in the column 4b implies that there exist one $4b^1$ digit set at that stage and the symbol $2 @ [2a \dot{+} a \leftarrow b \dot{+} a]$ means that two Radix-2 Carry Generators are used . In cases of ambiguities , more explicit symbols will be needed . For example , the symbol 2^2 in column 8b may mean either two $8b^1$ digit sets or one $8b^0$ plus one $8b^2$ digit sets . To avoid such ambiguities , we shall use I^1, I^1 for the former and I^0, I^2 for the latter case . The first row of each Information Loss Chart , beneath the column headings , contain values of the above quantities which are present on the input sideof an Adder structure to be designed . In the following rows , the values of parameters will vary (i.e. may increase ,decrease or remain unchanged depending on the particular fundamental structures being used in the row immediately above) . This continues until a pattern of weighted a and b digit sets exactly the same as that of the desired output is obtained . The Information Loss Chart so derived contains enough information that can be quickly translated into a block diagram layout .

Note that in the examples given below we only present for illustrative purposes what appear to be "reasonable" designs and do not claim that they are optimal ones .

Example I : Common digit set for the input and output operands is $n^7 = \{\bar{7}, \bar{6}, \ldots, 0, \ldots, 6, 7_2\}$. A Mythical Input digit set ,namely $d^2 = \{\bar{2}, \bar{I}, 0, I, 2\}$, is necessary in this case . After replacing each digit set by its Quasibinary representation , we obtain one as shown in the Information Loss Chart of Table I .

Note the '↵' marks in the Information Loss Chart which emphasise the fact that each Transfer In signal can only be used after its Transfer Out counterpart has been generated as indicated earlier in Fig. 4 .

Example 2 : Common digit set for the input and output operands is $o^8 = \{\bar{8}, \bar{7}, \ldots, 0, \ldots, 6, 7\}$. Again a Mythical Input digit set ,namely $c^1 = \{\bar{I}, 0, I, 2\}$, is necessary . After replacing each digit set by its Quasibinary representation , we obtain one decomposition as shown in the Information Loss Chart of Table II .

Note that ,in common electronic implementation of the fundamental structures , the FanOut Operators require no hardware and hence no Time Delay , its presence can be assimilated into the other fundamental structures , as illustrated in Fig.5 which occurs in the first stage of this example.

Beyond confirming the relations of Result 5 , the examples above (and others not presented here)lead to the following observations . As expected , due to the presence of an additional type of fundamental structures ,the Radix-m Carry Generator the decomposition process becomes more complicated . In the case of radix $r = 2^k$, a reasonable decomposition strategy that seems to have always worked is to combine the a and b digit sets of the same weights as quickly as they become available to generate higher weighted digit sets while at the same time avoiding the use of Converters as much as possible . This is because the available a and b digit sets are always in proper form for combinations using either Radix-2 Carry Generators or Generalised Half-Adders (except when all inputs are b digit sets where one is forced to use the Converters to keep the decomposition going) and because there are no ouput signals with weights other than 2^k , $k \geq I_k$. In contrast , in the case of radix $r = m^k$, $k \geq I$ with $m > 2$, the attention must focus on generating output signals with weights of the form m^k via Radix-m Carry Generators . Usually , one needs to rearrange the available digit sets before they are in proper form ready to be used as inputs to the Radix-m Carry Generators in order to generate the desired outputs . These rearrangements often require the use of FanOut Operators , of Converters and extra Time Delays . This means that it is not possible to state a general strategy for the case of radix $r = m^k, k \geq I$, $m > 2$. This also explains the fact that while the use of FanOut Operators may be avoided entirely (except for pathological cases of little practical interest) in the case of radix $r = 2^k, k \geq I$, such is not the case for radix $r = m^k$, $k \geq I$, $m > 2$,with Example 2 as an illustration

To complete the designs of the examples one has to fill in the implementation details of the fundamental structures . The logical designs of all versions of all variants of the last four types of fundamental structures for radix $r = I0^k$, $k \geq I$, are given in [1] . It remains to consider the logical designs of the Radix-5 Carry Generators. There exist I4 variants of this type of fundamental structures , each one corres ponds to a value of the offset . Because each such structure involves 2 b digit sets , each one can independently assume one of the 9 formats mentioned earlier, 8I versions of logical designs for each variant must be considered .Because the logical designs structures which are negative of one another are identical or can be easily deduced from one another ,we need only concern with 7 different variants and hence $(7 * 8I) = 567$ different logical design problems, each one with 4 binary va-

riables as inputs and 4 variables as outputs . Again space limitations prevent discussions of these logical designs here. Reference 9 contains two typical designs as examples . When the need arises the designer can carry out similar designs which are not difficult in nature .

Based on our limited designs experiences , we cannot ascertain which format for the b digit sets in the Radix-m Carry Generators would results in better designs. However, for practical cases that we have come across the number of Radix-m Carry Generators required is uasually small in comparison to the numbers of Radix-2 Carry Generators , and Generalised Half-Adders . As a consequence , the observation[I] that the use of Format #2 or Format #3 would result in overall better designs in terms of gate counts could be expected to hold true for the case of radix $r = m^k$, $k \geq I$, $m > 2$ as in the case of radix $r = 2^k$, $k \geq I$.

5. Conclusion

The results of Robertson [I] concerning a systematic approach to the design of Adder/Subtracter structures of radix $r = 2^k$, $k \geq I$ have been generalised to cover all structures of radix $r = m^k$, $k \geq I$, $m \geq 2$. The use of the so-called Quasibinary representations for all digit sets involved in these structures is shown to play an important role here as in the earlier case . In addition to the four types of fundamental structures already discussed in [I] ,only one new type ,called the Radix-m Carry Generator , is needed .

There exist other small differences between the case of radix $r = 2^k$, $k \geq I$ and the case of radix $r = m^k$, $k \geq I$, $m > 2$ .Some examples involving Decimal Adder structures illustrate these differences . The observations made for these examples hold true for other values of $m > 2$ as well .As in the case of radix $r = 2^k$, $k \geq I$, it is possible to derive a set of relations connecting the numbers of different types of fundamental structures being used in any design and the parameters of that design problem, which may be of use to the designer .

As can be seen several issues concerning the application of the approach to practical problems are left untouched .Issues such as the choice of a particular representation for the Mythical Input , the compromise between hardware complexity and speed ... must be dealt with by the designer of these problems .

ACKNOWLEDGEMENTS

To Professor James E. Robertson for his support and guidance and to the National Science Foundation for its financial support (NSF Grant MCS 80-00058)

BIBLIOGRAPHY

I. ROBERTSON J.E. " A Systematic Approach to the Design of Structures for Arithmetic" Tech. Rept. UIUCDCS-R-80-I004 ,Dept. of Computer Science ,University of Illinois at Urbana-Champaign, Jan. I98I .

2. ATKINS D.E. " Introduction to the role of Redundancy in Computer Arithmetic ", Computer , Jun. I975 .

3. ATKINS D.E. " Design of the Arithmetic Unit of ILLIAC III : Use of Redundancy and Higher Radix Method ",IEEE Trans. on Computers , vol. C-I9 , N. 8 , Aug. I970 .

4. BOROVEC R.T. " The Logical Design of a Class of Limited-Carry-Propagation Adders Tech. Rept. 275 ,Dept. of Computer Science, University of Illinois at Urbana-Champaign, I968 .

5. CHOW C.Y. and ROBERTSON J.E. "Logical Design of a Redundant Binary Adder " Proc. of Second Symp. on Computer Arithmetic , I978 .

6. METZE G.A. and ROBERTSON J.E. " Elimination of Carry Propagation in Digital Computres " Proc. of Int. Conf. on Information Processing , Paris ,France ,Jun. I959 .

7. ROBERTSON J.E. Lecture Notes for CS 364 ,Dept. of Computer Science , University of Illinois at Urbana-Champaign ,Aug. I979.

8. ROHATSCH F.A. " A Study of Transformation Applicable to the Development of Limited-Carry-Borrow-Propagation Adders " Tech. Rept. 226 , Dept. of Computer Science ,University of Illinois at Urbana-Champaign , Jun.I967 .

9. NGUYEN D.D. "A Systematic Approach to the Design of Structures for Addition and Subtraction _ Case of Radix $r = m^k$." Tech. Rept. UIUCDCS-R-80-I049 , Dept. of Computer Science , University of Illinois at Urbana Champaign ,Dec. I980 .

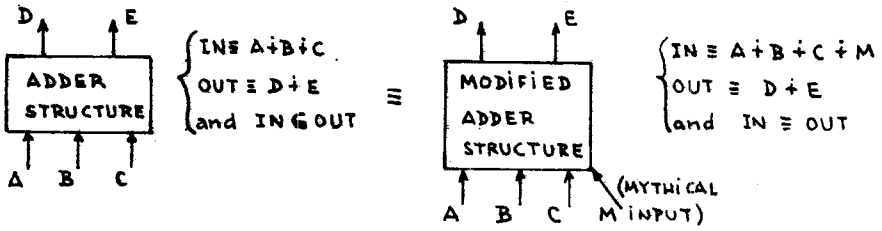

FIG. 1 - USE OF MYTHICAL INPUT DIGIT SET

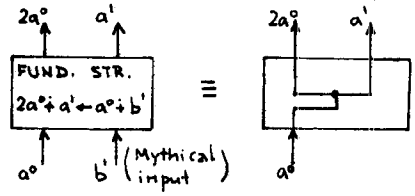

FIG. 2 - FUNDAMENTAL STRUCTURE WITH A MYTHICAL INPUT AND CORRESPONDING REDUCED STRUCTURE.

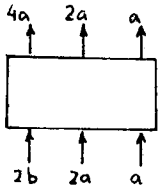

FIG. 3.a

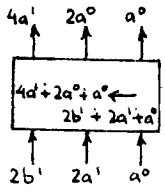

FIG. 3.b

EXAMPLE OF A FUNDAMENTAL STRUCTURE AND ONE OF ITS VARIANT.

TABLE I

λ β α	10 b	10 a	8 b	8 a	4 b	4 a	2 b	2 a	1 b	1 a	Numbers and types of fundamental structures
14 6 2					1', 1'		1', 1'	2'	1', 1'		1@[a+a ← b] ; 2@[2a+a ← b+a]
					1', 1'	2'		2°	1'	2'	1@[b ← a+a] ; 3@[2a+a ← b+a]
				2^2		2°	1°	1'		2°	2@[b ← a+a] ; 1@[2a+a ← b+a]
				2^2	1°	1°		1'	1°		1@[5a+2a+b ← 4a+2b+a]
8 2 4		1'		1'		2°	1°		1°		
8 2 4				1'		2°	1°		1°	1'	↵ 1@[b ← a+a] ; 1@[2a+a ← b+a]
				1'	1°		1°	1°		1'	1@[2a+a ← b+a]
				1'	1°	1°		1°		1'	1@[5a+2a+b ← 4a+2b+a]
6 1 4		1°				2'	1'			1'	
6 1 4						2'	1'			2'	↵ 2@[b ← a+a]
					1'		1'		1'		

SUMMARY: $\Delta\lambda = 8$, $\Delta\beta = 3$, $\Delta\alpha = 2$ and $\sigma_{10} = 2$.
$z_5 = 2$, $z = 8$, $y = 4$, $x = 1$ and $u = 0$. TIME DELAY = 8 UNITS.

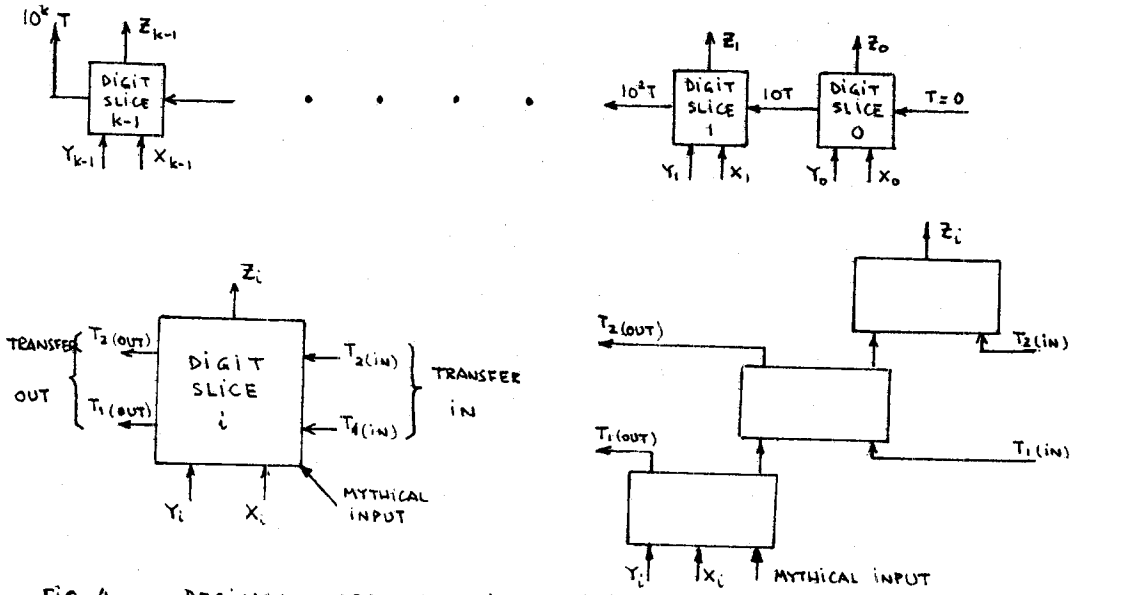

FIG. 4 - DECIMAL ADDER WITH LIMITED-CARRY-PROPAGATION PROPERTY.

TABLE II

λ β α	10 b	10 a	8 b	8 a	4 b	4 a	2 b	2 a	1 b	1 a	Numbers and types of fundamental structures
10 0 10				2^2		2^0		3^0		3^1	{1@ FANOUT; 4@[b←a+a]
				1^1	$1^1, 1^1$		1^0	1^0	1^0	1^1	{1@ [2a+a ← b+a] 1@[5a+2a+b←4a+2b+a]
10 3 4		1^1			1^1	1^0	2^1	1^0		1^1	
10 3 4					1^1	1^0	2^1	1^0		2^2	↵ {1@ [b ← a+a] 2@ [2a+a ← b+a]
				1^0		2^1	1^0	1^1	1^2		{1@ [b ← a+a] 1@ [2a+a ← b+a]
				1^0	1^1	1^0		1^1	1^2		{1@ [5a+2a+b←4a+2b+a]
7 2 3		1^0				2^1	1^1		1^2		
7 2 3						2^1	1^1		1^2	1^0	↵ {1@[b ← a+a] 1@[2a+a ← b+a]
					1^1		1^1	1^1		1^0	{1@ [2a+a ← b+a]
					1^1	1^1		1^0		1^0	{1@ [2a+a ← b+a]
				1^1		1^0		1^0		1^0	

SUMMARY: $\Delta\lambda = 6$, $\Delta\beta = 0$, $\Delta\alpha = 6$, $\sigma_{10} = 2$.
$z_5 = 2$, $z = 7$, $y = 7$, $x = 0$, $u = 1$. TIME DELAY = 8 UNITS.