

## LOGICAL DESIGN OF A REDUNDANT BINARY ADDER

Catherine Y. Chow, James E. Robertson

University of Illinois, Urbana, Illinois

### Abstract

This paper investigates the logical design of a redundant binary adder with two input digits and one output digit, all in the digit set  $\{\bar{1}, 0, 1\}$ . Redundant binary arithmetic structures in which all digit sets are  $\{\bar{1}, 0, 1\}$  were first discussed by Avizienis in 1961. Borovec studied the logical design of a class of such binary adders and subtractors in 1968. At that time, a variation of the adder/subtractor was overlooked. This paper studies the logical design of this variation. The sum digit is still a function only of the digits in three adjacent digital positions of the operands. "Coupled don't cares" are encountered, but have not introduced too much difficulty. The nine distinct formats (under permutation and negation) of representing three values with two bits given by Robertson are used. The simplest adder/subtractor designs from this variation are less complex than the simplest designs previously known.

### Background

This paper investigates the logical design of a redundant binary adder with two inputs and one output, all in the digit set  $\{\bar{1}, 0, 1\}$ .

The available hardware technologies have certainly made the implementation of redundant arithmetic algorithms practical. The redundant binary adder is a basic building unit in redundant arithmetic structures whether they perform fast digit parallel operations without propagating transfers or most-significant-digit-first digit serial operations<sup>1,2</sup>. The redundant binary adder is heavily used in Goyal's LSI implementable arithmetic unit<sup>1</sup>. Goyal also showed that radix-2<sup>k</sup> operations can be easily implemented with redundant binary adders. Although LSI technologies have made logic design minimization less urgent, the redundant binary adder which may be used hundred of times even in a modest

arithmetic unit deserves to be studied in detail. With the advent of digit pipelining algorithms which inevitably require a redundant number representation, the authors are convinced that the redundant binary adder will be a fundamental structure in the future.

A redundant arithmetic structure in which all digit sets are  $\{\bar{1}, 0, 1\}$  was first discussed by Avizienis.<sup>3</sup> He showed that the requirements of a totally parallel operation and a unique representation of zero dictate a three-level structure for the signed-digit binary adder/subtractor. The result digit depends only on digits in three adjacent digital positions of the operands. Borovec investigated the logical design of this structure.<sup>4</sup> One of his two approaches (Figure 1) is to split the sum of the operand digits in the  $i^{\text{th}}$  position,  $\ell_i^*$  and  $k_i^*$ , into  $a_{i-1}^*$  and  $m_i$  such that

$$\ell_i^* + k_i^* = 2 \cdot a_{i-1}^* + m_i$$

and input these to adder/subtractor structures which operate on one redundant digit and one nonredundant digit. (An asterisk beside a symbol denotes a redundant binary digit.  $(i-1)$  is the position of the left neighboring unit.) His second approach consists of combining the two lower levels so that the format of  $a_i^*$  need not be considered (Figure 2). This paper studies the structures resulting from having different outputs from the lowest level. The sum of the pair of operand digits in the  $i^{\text{th}}$  position is expressed as  $(2 \cdot m_{i-1}^* + a_i^*)$  as illustrated in Figure 3.

### The Adder Structure

Since it is not trivial to negate a number expressed in the normalized redundant binary digit set, the operands digits  $\ell_i^*$  and  $k_i^*$  are restricted to the symmetric digit set  $\{\bar{1}, 0, 1\}$ . The structure shown

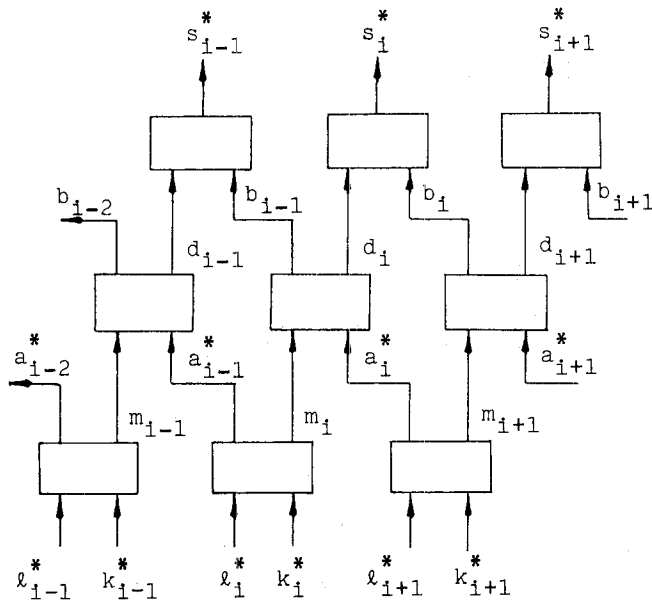


Figure 1 Redundant binary adder/subtractor structure with three separate levels

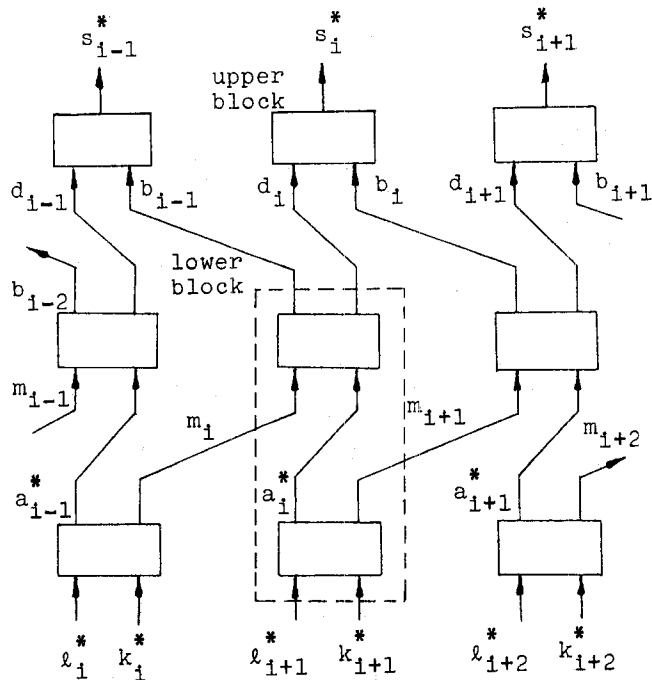


Figure 2 Combining the two lower levels

in Figure 3 can again be built either in three separate levels or with the two lower levels combined. However, the arrangement with three separate levels is not very useful in this case because the valid groups of digit sets for  $m_i$  and  $a_i^*$  are not practical. The sum of  $l_i^*$  and  $k_i^*$  equals  $(2 \cdot m_{i-1} + a_i^*)$ . Hence, the digit sets for  $m_{i-1}$  and  $a_i^*$  must be from one of the following combinations:

1.  $m_{i-1} \in \{\bar{1}, 0\}$ ,  $a_i^* \in \{0, 1, 2\}$ ,
2.  $m_{i-1} \in \{0, 1\}$ ,  $a_i^* \in \{\bar{2}, \bar{1}, 0\}$ , or
3.  $2 \cdot m_{i-1} \in \{\bar{1}, 1\}$ ,  $a_i^* \in \{\bar{1}, 0, 1\}$ .

For the design with three separate levels, one would like to use the same digit set for the redundant transfer digit, the two operand digits and the sum digit

( $a_i^*$ ,  $l_i^*$ ,  $k_i^*$ , and  $s_i^*$  respectively) and to have  $m_{i-1}$  as a conventional binary digit.

Failing to satisfy these, the three-separate-levels arrangement needs no further attention.

Combining the two lower levels does result in relatively simple logic designs (Figure 4). For the symmetric adder, the algebraic relationships are

1.  $d_i + b_i = s_i$  for the upper block, and
2.  $l_i^* + k_i^* + m_i = 2 \cdot m_{i-1} + 2 \cdot b_{i-1} + d_i$  for the lower block.

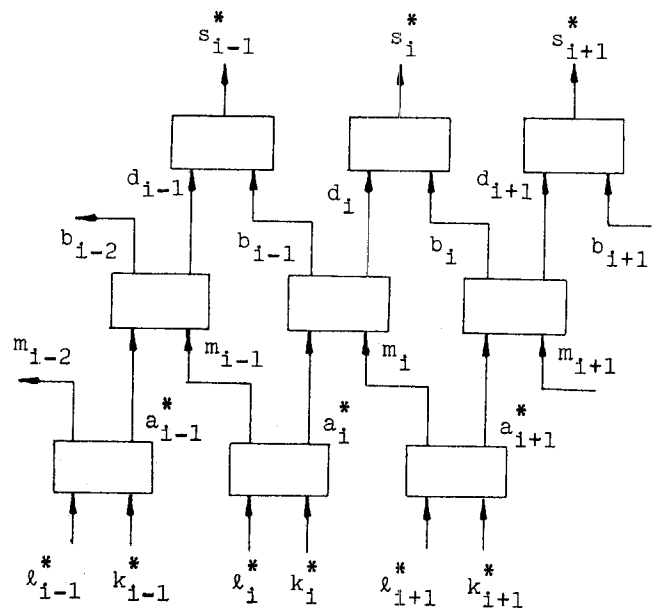


Figure 3 Variation of the three level structure

In spite of the above equation,  $m_{i-1}$  can be made independent of  $m_i$ . The final sum digit  $s_i^*$  is still a function only of the digits in three adjacent digital positions of the operands. The chosen sets for  $m_i$ ,  $b_i$ ,  $d_i$  are  $\{0, 1\}$ ,  $\{\bar{1}, 0\}$ ,  $\{0, 1\}$  respectively. Before actually designing the logic circuit, one has to fix the representation of the redundant binary digit. Robertson<sup>5</sup> has shown that there exist only nine distinct ways, under permutation and negation, of representing three values with two bits. The nine formats are shown in Table 1. Since it is sometimes necessary to feed the output from an adder back as an operand, the sum digit  $s_i^*$  should have the same format as the operand digits  $\ell_i^*$  and  $k_i^*$ .

#### Details

Some rather interesting details are encountered in the logic designing of this adder. Whenever  $(m_i + \ell_i^* + k_i^*)$  is algebraically 0 or 1,  $(-b_{i-1})$  and  $m_{i-1}$  can be both 0 or both 1. This results in  $b_{i-1}$  and  $m_{i-1}$  having "coupled don't cares" in the truth tables.<sup>4</sup> As an example, Table 2

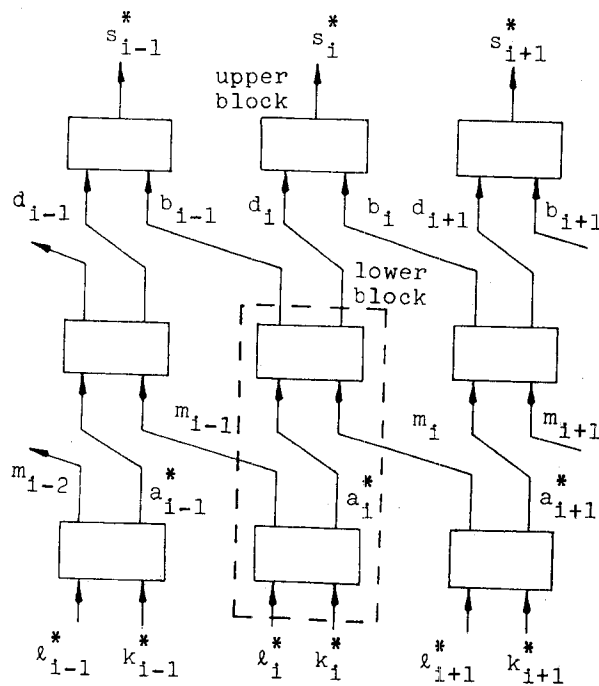


Figure 4 Combining the two lower levels of Figure 3

shows the truth table for the lower block of the adder using format 2 from Table 1.  $\ell_i^*$ ,  $k_i^*$  and  $s_i^*$  are represented by pairs of bits  $(\lambda_i, \ell_i)$ ,  $(\kappa_i, k_i)$  and  $(\sigma_i, s_i)$  respectively. The horizontal pairs of c's

$s_i^*$		PN distinct formats								
$\sigma_i$	$s_i$	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	d	0	0	1	$\bar{1}$
0	1	1	1	1	1	1	1	1	1	1
1	0	$\bar{1}$	d	0	$\bar{1}$	0	1	$\bar{1}$	0	0
1	1	0	$\bar{1}$	$\bar{1}$	d	$\bar{1}$	$\bar{1}$	$\bar{1}$	$\bar{1}$	$\bar{1}$

Table 1 Nine PN (permutation-negation) distinct formats of representing a redundant binary digit with two bits

	$m_i$	$\lambda_i$	$\ell_i$	$\kappa_i$	$k_i$	value	$m_{i-1}$	$b_{i-1}$	$d_i$
0	0	0	0	0	0	0	c	c	0
1	0	0	0	0	1	1	c(1)	c(1)	1
2	0	0	0	1	0	d	d	d	d
3	0	0	0	1	1	$\bar{1}$	0	1	1
4	0	0	1	0	0	1	c(1)	c(1)	1
5	0	0	1	0	1	2	1	0	0
6	0	0	1	1	0	d	d	d	d
7	0	0	1	1	1	0	c	c	0
8	0	1	0	0	0	d	d	d	d
9	0	1	0	0	1	d	d	d	d
10	0	1	0	1	0	d	d	d	d
11	0	1	0	1	1	d	d	d	d
12	0	1	1	0	0	$\bar{1}$	0	1	1
13	0	1	1	0	1	0	c	c	0
14	0	1	1	1	0	d	d	d	d
15	0	1	1	1	1	$\bar{2}$	0	1	0
16	1	0	0	0	0	1	c	c	1
17	1	0	0	0	1	2	1	0	0
18	1	0	0	1	0	d	d	d	d
19	1	0	0	1	1	0	c(0)	c(0)	0
20	1	0	1	0	0	2	1	0	0
21	1	0	1	0	1	3	1	0	1
22	1	0	1	1	0	d	d	d	d
23	1	0	1	1	1	1	c	c	1
24	1	1	0	0	0	d	d	d	d
25	1	1	0	0	1	d	d	d	d
26	1	1	0	1	0	d	d	d	d
27	1	1	0	1	1	d	d	d	d
28	1	1	1	0	0	0	c(0)	c(0)	0
29	1	1	1	0	1	1	c	c	1
30	1	1	1	1	0	d	d	d	d
31	1	1	1	1	1	$\bar{1}$	0	1	1

Table 2 Truth table for the lower block of format 2

across  $m_{i-1}$  and  $b_{i-1}$  are the "coupled don't cares". A horizontal pair of c's can be both 0 or both 1.

The complexity introduced by the "coupled don't cares" is greatly reduced by the important constraint that the transfers  $m_{i-1}$  and  $b_{i-1}$  be non-propagating. This means that  $m_{i-1}$  must be independent of  $m_i$  and fixes the values of most of the "coupled don't cares". In Table 2, the "coupled don't cares" for minterms 1, 4, 19 and 28 must have the values 1, 1, 0, and 0 respectively because the upper 16 function values of  $m_{i-1}$  must match the lower 16 function values of  $m_{i-1}$ . Figure 5 shows the Karnaugh maps for  $m_{i-1}$  from Table 2. Besides the coupled relationship between  $m_{i-1}$  and  $b_{i-1}$ , the "coupled don't cares" are also coupled across the maps corresponding to  $m_i = 0$  and  $m_i = 1$  as indicated by the dotted lines in Figure 5. For this particular format, there remain only 8 possible assignments of the values of  $m_{i-1}$ . The simplest design for this format is given in Table 3.

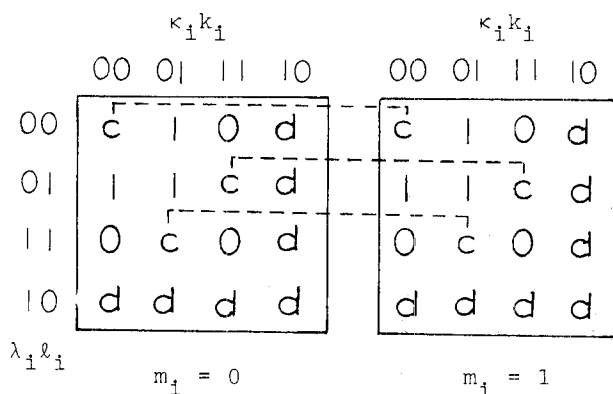


Figure 5 Karnaugh maps for  $m_{i-1}$  for the adder with format 2

The other formats for the redundant binary digit are similarly studied. The criterion used for choosing a design for each format is that the boolean functions yield a minimal total number of literals when they are expressed as sums of products or products of sums.

If the chosen digit sets for  $m_i$ ,  $b_i$  and  $d_i$  are negated, the resulting combination of digit sets is still valid. However, there is no need to investigate the logic designs of the negated digit sets

because they are similar in complexity to the designs of the original digit sets. For each of the nine formats, the truth table for the negated digit sets can be obtained from the truth table of the chosen digit sets by changing  $m_i$  to  $\bar{m}_i$  and switching the output functions  $b_{i-1}$  and  $m_{i-1}$ .

Table 4 shows the designs for the symmetric subtracter. For the symmetric subtracter, the algebraic relationship for the lower block of Figure 4 is

$$m_i + l_i^* - k_i^* = 2 \cdot m_{i-1} + 2 \cdot b_{i-1} + d_i.$$

The upper block is the same as that of the adder. The nine formats are again used to yield the designs shown in Table 4.

The simplest designs both for the adder and the subtracter are those corresponding to formats 2, 4 and 5. These designs are simpler than the best designs given in reference 4 (the appendix).

1.  $d_i = m_i \oplus \lambda_i \oplus l_i \oplus \kappa_i \oplus k_i$   
 $m_{i-1} = \bar{\lambda}_i \bar{\kappa}_i \vee l_i k_i$   
 $b_{i-1} = \lambda_i \kappa_i (\bar{m}_i \vee \bar{l}_i \bar{k}_i \vee l_i k_i) \vee \bar{m}_i \bar{l}_i \bar{k}_i \vee \bar{m}_i l_i k_i (\kappa_i \vee \lambda_i) \vee \bar{\lambda}_i \bar{\kappa}_i (\bar{m}_i \bar{l}_i \vee \bar{m}_i \bar{k}_i \vee \bar{l}_i \bar{k}_i)$   
 $\sigma_i = b_i$   
 $s_i = d_i$
2.  $d_i = m_i \oplus l_i \oplus k_i$   
 $m_{i-1} = \bar{\lambda}_i \bar{\kappa}_i (l_i \vee k_i)$   
 $b_{i-1} = \bar{m}_i \bar{l}_i k_i \vee \bar{m}_i l_i \bar{k}_i \vee \lambda_i \kappa_i$   
 $\sigma_i = \bar{d}_i b_i$   
 $s_i = d_i \oplus b_i$
3.  $d_i = m_i \oplus l_i \oplus k_i$   
 $m_{i-1} = \bar{\kappa}_i k_i \vee \bar{\lambda}_i l_i \bar{k}_i$   
 $b_{i-1} = \bar{m}_i l_i \bar{k}_i \vee \bar{m}_i \bar{l}_i k_i \vee \lambda_i l_i k_i$   
 $\sigma_i = b_i$   
 $s_i = d_i \oplus b_i$
4.  $d_i = m_i \oplus \bar{\lambda}_i \bar{l}_i \oplus \bar{\kappa}_i \bar{k}_i$   
 $m_{i-1} = \bar{\lambda}_i \bar{\kappa}_i$   
 $b_{i-1} = \bar{m}_i \bar{\lambda}_i \bar{l}_i \vee \lambda_i \kappa_i \vee \bar{\kappa}_i \bar{k}_i (\bar{m}_i \vee \bar{\lambda}_i \bar{l}_i)$   
 $\sigma_i = \bar{d}_i b_i$   
 $s_i = d_i b_i$

Table 3(a) Boolean equations for symmetric adders with formats 1 to 4

$$\begin{aligned}
5. \quad d_i &= m_i \oplus \ell_i \oplus k_i \\
m_{i-1} &= \bar{\lambda}_i \bar{k}_i \vee \bar{\kappa}_i \\
b_{i-1} &= \lambda_i k_i (\bar{m}_i \vee \ell_i) \vee \bar{m}_i \ell_i \bar{k}_i \\
\sigma_i &= \bar{d}_i \vee b_i \\
s_i &= d_i \oplus b_i \\
\\
6. \quad d_i &= m_i \oplus \bar{\lambda}_i \bar{\ell}_i \oplus \bar{\kappa}_i \bar{k}_i \\
m_{i-1} &= \bar{\lambda}_i \bar{k}_i \vee \bar{\ell}_i \bar{k}_i \vee \kappa_i \bar{k}_i \\
b_{i-1} &= \bar{\lambda}_i \bar{\ell}_i (\bar{m}_i \vee \bar{\kappa}_i \bar{k}_i) \vee \bar{m}_i \bar{\kappa}_i \bar{k}_i \\
&\quad \vee \ell_i \lambda_i \kappa_i \\
\sigma_i &= b_i \bar{d}_i \\
s_i &= d_i \oplus b_i \\
\\
7. \quad d_i &= m_i \oplus \bar{\lambda}_i \bar{\ell}_i \oplus \bar{\kappa}_i \bar{k}_i \\
m_{i-1} &= \bar{\lambda}_i \bar{k}_i \\
b_{i-1} &= \bar{\lambda}_i \bar{\ell}_i (\bar{m}_i \vee \bar{\kappa}_i \bar{k}_i) \vee \lambda_i \kappa_i \vee \bar{m}_i \bar{\kappa}_i \bar{k}_i \\
\sigma_i &= \bar{d}_i b_i \\
s_i &= d_i \bar{b}_i \\
\\
8. \quad d_i &= \lambda_i \bar{\ell}_i \oplus \kappa_i \bar{k}_i \oplus m_i \\
m_{i-1} &= \bar{\lambda}_i \vee \bar{\kappa}_i \\
b_{i-1} &= (\lambda_i \vee \kappa_i) (\bar{\lambda}_i \vee \ell_i \vee \bar{\kappa}_i \vee k_i) \\
&\quad \cdot (\bar{m}_i \vee \bar{\kappa}_i \vee k_i) (\bar{m}_i \vee \bar{\lambda}_i \vee \ell_i) \\
\sigma_i &= \bar{d}_i \vee b_i \\
s_i &= \bar{d}_i b_i \\
\\
9. \quad d_i &= \lambda_i \bar{\ell}_i \oplus \kappa_i \bar{k}_i \oplus m_i \\
m_{i-1} &= \bar{\lambda}_i \ell_i \vee \bar{\ell}_i \kappa_i \bar{k}_i \\
b_{i-1} &= (\bar{\ell}_i \vee \kappa_i \vee \bar{k}_i) (\bar{\lambda}_i \vee \ell_i \vee \bar{\kappa}_i \vee k_i) \\
&\quad \cdot (\bar{m}_i \vee \bar{\kappa}_i \vee k_i) (\bar{m}_i \vee \bar{\lambda}_i \vee \ell_i) \\
\sigma_i &= \bar{d}_i \vee b_i \\
s_i &= d_i \oplus b_i
\end{aligned}$$

Table 3(b) Boolean equations for symmetric adders with formats 5 to 9

$\oplus$  denotes "exclusive or";  
 $\vee$  denotes "or";  
juxtaposition or  $\cdot$  denotes "and".

$$\begin{aligned}
1. \quad d_i &= m_i \oplus \lambda_i \oplus \ell_i \oplus \kappa_i \oplus k_i \\
m_{i-1} &= (\ell_i \vee \kappa_i) (\bar{\lambda}_i \vee \bar{k}_i) \\
b_{i-1} &= \bar{m}_i (\lambda_i \oplus k_i) \vee \lambda_i \bar{\ell}_i \bar{k}_i k_i \\
&\quad \vee (\ell_i \oplus \kappa_i) ((\lambda_i \oplus k_i) \vee \bar{m}_i) \\
\sigma_i &= b_i \\
s_i &= d_i \\
\\
2. \quad d_i &= m_i \oplus \ell_i \oplus k_i \\
m_{i-1} &= \kappa_i \vee \bar{\lambda}_i \ell_i \bar{k}_i \\
b_{i-1} &= \bar{m}_i \bar{\ell}_i k_i \vee \bar{m}_i \ell_i \bar{k}_i \vee \lambda_i k_i \\
\sigma_i &= \bar{d}_i b_i \\
s_i &= d_i \oplus b_i \\
\\
3. \quad d_i &= m_i \oplus \ell_i \oplus k_i \\
m_{i-1} &= \bar{\lambda}_i \ell_i \bar{k}_i \vee \kappa_i k_i \\
b_{i-1} &= \bar{m}_i \bar{\ell}_i k_i \vee \bar{m}_i \ell_i \bar{k}_i \vee \lambda_i \ell_i k_i \\
\sigma_i &= b_i \\
s_i &= d_i \oplus b_i \\
\\
4. \quad d_i &= m_i \oplus \bar{\lambda}_i \bar{\ell}_i \oplus \bar{\kappa}_i \bar{k}_i \\
m_{i-1} &= \bar{\lambda}_i \vee \bar{k}_i \\
b_{i-1} &= \bar{\kappa}_i \bar{k}_i (\bar{m}_i \vee \bar{\lambda}_i \bar{\ell}_i) \vee \bar{m}_i \bar{\lambda}_i \bar{\ell}_i \vee \lambda_i k_i \\
\sigma_i &= \bar{d}_i b_i \\
s_i &= d_i \bar{b}_i \\
\\
5. \quad d_i &= m_i \oplus \ell_i \oplus k_i \\
m_{i-1} &= \bar{\lambda}_i \vee \bar{\ell}_i \kappa_i k_i \\
b_{i-1} &= \bar{m}_i \bar{\ell}_i k_i \vee \bar{m}_i \ell_i \bar{k}_i \vee \ell_i \bar{k}_i \\
\sigma_i &= \bar{d}_i \vee b_i \\
s_i &= d_i \vee b_i \\
\\
6. \quad d_i &= m_i \oplus \bar{\lambda}_i \bar{\ell}_i \oplus \bar{\kappa}_i \bar{k}_i \\
m_{i-1} &= \kappa_i k_i \vee \bar{\lambda}_i \bar{\kappa}_i \bar{k}_i \vee \bar{\ell}_i \bar{\kappa}_i \bar{k}_i \\
b_{i-1} &= \bar{m}_i \bar{\lambda}_i \bar{\ell}_i \vee \bar{m}_i \bar{\kappa}_i \bar{k}_i \vee \bar{\lambda}_i \bar{\ell}_i \bar{\kappa}_i \bar{k}_i \\
&\quad \vee \lambda_i \ell_i \kappa_i \vee \lambda_i \ell_i k_i \\
\sigma_i &= b_i \bar{d}_i \\
s_i &= d_i \oplus b_i
\end{aligned}$$

Table 4(a) Boolean equations for symmetric subtractors with formats 1 to 6

$$\begin{aligned}
7. \quad d_i &= m_i \oplus \bar{\lambda}_i \bar{\ell}_i \oplus \bar{\kappa}_i \bar{k}_i \\
m_{i-1} &= \bar{\lambda}_i \bar{k}_i \vee \kappa_i k_i \\
b_{i-1} &= \bar{m}_i \bar{\lambda}_i \bar{\ell}_i \vee \bar{m}_i \bar{\kappa}_i \bar{k}_i \vee \lambda_i k_i \\
&\quad \vee \bar{\lambda}_i \bar{\ell}_i \bar{\kappa}_i \bar{k}_i \\
\sigma_i &= b_i \bar{d}_i \\
s_i &= d_i \bar{b}_i \\
8. \quad d_i &= m_i \oplus \lambda_i \bar{\ell}_i \oplus \kappa_i \bar{k}_i \\
m_{i-1} &= \kappa_i (\bar{\lambda}_i \vee \bar{\ell}_i) \\
b_{i-1} &= m_i \lambda_i \bar{\ell}_i \vee \lambda_i \bar{\ell}_i \kappa_i \bar{k}_i \vee \bar{m}_i \kappa_i \bar{k}_i \\
&\quad \vee \lambda_i \bar{\ell}_i \bar{\kappa}_i \\
\sigma_i &= \bar{d}_i \vee b_i \\
s_i &= d_i \oplus b_i \\
9. \quad d_i &= m_i \oplus \lambda_i \bar{\ell}_i \oplus \kappa_i \bar{k}_i \\
m_{i-1} &= \bar{\lambda}_i \bar{\ell}_i \vee \lambda_i \bar{\ell}_i \bar{k}_i \vee \lambda_i \bar{\ell}_i \kappa_i \\
b_{i-1} &= \bar{m}_i \kappa_i \bar{k}_i \vee \bar{m}_i \lambda_i \bar{\ell}_i \vee \bar{\ell}_i \bar{\kappa}_i k_i \\
&\quad \vee \lambda_i \bar{\ell}_i \kappa_i \bar{k}_i \vee \lambda_i \bar{\ell}_i k_i \\
\sigma_i &= \bar{d}_i \vee b_i \\
s_i &= d_i \oplus b_i
\end{aligned}$$

Table 4(b) Boolean equations for symmetric subtractors with formats 7 to 9

#### Appendix

Tables 1 and 2 contain the lower block designs given by Borovec in reference 4. As shown in Figure 2, the inputs to the lower block of the  $(i)^{th}$  unit are  $\ell_i^* = (\lambda_i, l_i)$ ,  $k_i^* = (\kappa_i, k_i)$  and  $m_{i-1}$ ; the outputs are  $m_i$ ,  $d_{i-1}$  and  $b_{i-2}$ . The output value  $m_i + 2 \cdot d_{i-1} + 4 \cdot b_{i-2}$  is unique in this arrangement.

For the symmetric adder designs the digit sets for  $b_i$ ,  $m_i$ , and  $d_i$  are  $\{0, 1\}$ ,  $\{0, 1\}$  and  $\{\bar{1}, 0\}$  respectively. For the symmetric subtractor, the corresponding sets are  $\{\bar{1}, 0\}$ ,  $\{\bar{1}, 0\}$  and  $\{0, 1\}$ .

$$\begin{aligned}
1. \quad m_i &= \kappa_i \oplus \lambda_i \oplus l_i \oplus k_i \\
d_{i-1} &= m_{i-1} \oplus (\bar{\kappa}_i \bar{k}_i \vee \lambda_i \bar{l}_i \vee \bar{\kappa}_i k_i (\bar{\lambda}_i \vee l_i) \\
&\quad \vee \bar{\lambda}_i l_i (\bar{\kappa}_i \vee k_i)) \\
b_{i-2} &= \kappa_i \bar{k}_i \lambda_i l_i \vee m_{i-1} (\lambda_i l_i \vee \kappa_i \bar{k}_i \vee \bar{l}_i \bar{k}_i \\
&\quad \vee \kappa_i \bar{\lambda}_i) \\
2. \quad m_i &= \kappa_i \oplus l_i \\
d_{i-1} &= \kappa_i \lambda_i \vee \bar{m}_{i-1} \bar{\kappa}_i \bar{\lambda}_i (\kappa_i \vee l_i) \\
&\quad \vee m_{i-1} (\lambda_i \vee \bar{l}_i \kappa_i \vee \bar{l}_i \bar{k}_i) \\
b_{i-2} &= \kappa_i \lambda_i \vee m_{i-1} (\lambda_i \vee \kappa_i \vee \bar{l}_i \bar{k}_i) \\
3. \quad m_i &= \kappa_i \oplus l_i \\
d_{i-1} &= m_{i-1} \oplus (\bar{\kappa}_i \bar{\lambda}_i l_i \vee \bar{k}_i \bar{\lambda}_i l_i \vee \bar{\kappa}_i k_i \bar{l}_i \\
&\quad \vee \kappa_i k_i \lambda_i l_i) \\
b_{i-2} &= \kappa_i k_i \lambda_i l_i \vee m_{i-1} (\kappa_i k_i \vee \lambda_i l_i \vee \bar{k}_i \bar{l}_i) \\
4. \quad m_i &= \bar{\kappa}_i \bar{k}_i \oplus \bar{\lambda}_i \bar{l}_i \\
d_{i-1} &= \bar{m}_{i-1} (\bar{\kappa}_i \bar{l}_i \vee \kappa_i \bar{\lambda}_i \vee \kappa_i \lambda_i) \\
&\quad \vee m_{i-1} (\kappa_i \oplus \lambda_i) \\
b_{i-2} &= \kappa_i \lambda_i \vee m_{i-1} (\kappa_i \vee \lambda_i \vee \bar{k}_i \bar{l}_i) \\
5. \quad m_i &= \kappa_i \oplus l_i \\
d_{i-1} &= \bar{m}_{i-1} (\kappa_i k_i \lambda_i l_i \vee \bar{\kappa}_i \bar{\lambda}_i \vee \bar{\kappa}_i \bar{l}_i \vee \bar{k}_i \bar{\lambda}_i) \\
&\quad \vee m_{i-1} (\kappa_i \bar{l}_i \vee \bar{\kappa}_i \lambda_i l_i \vee \kappa_i k_i \bar{l}_i \\
&\quad \vee \kappa_i k_i \bar{\lambda}_i) \\
b_{i-2} &= \kappa_i k_i \lambda_i l_i \vee m_{i-1} (\kappa_i k_i \vee \lambda_i l_i \vee \lambda_i \kappa_i) \\
6. \quad m_i &= \bar{\kappa}_i \bar{k}_i \oplus \bar{\lambda}_i \bar{l}_i \\
d_{i-1} &= m_{i-1} \oplus ((\bar{\lambda}_i \vee \bar{l}_i) (\kappa_i \oplus k_i) \\
&\quad \vee \bar{k}_i (\lambda_i \oplus l_i) \vee \kappa_i k_i \lambda_i l_i) \\
b_{i-2} &= \kappa_i k_i \lambda_i l_i \vee m_{i-1} (\kappa_i k_i \vee \lambda_i l_i \\
&\quad \vee \bar{\kappa}_i \bar{k}_i \bar{\lambda}_i \bar{l}_i) \\
7. \quad m_i &= \kappa_i \bar{\lambda}_i \bar{l}_i \vee \bar{\kappa}_i \bar{k}_i \bar{l}_i \\
d_{i-1} &= \bar{m}_{i-1} (\bar{\kappa}_i \bar{k}_i \bar{\lambda}_i \vee \bar{\kappa}_i \bar{\lambda}_i l_i \vee \kappa_i \lambda_i) \\
&\quad \vee m_{i-1} ((\kappa_i \oplus \lambda_i) \vee \kappa_i \bar{k}_i \bar{l}_i) \\
b_{i-2} &= \kappa_i \lambda_i \vee m_{i-1} (\kappa_i \vee \lambda_i \vee \bar{k}_i \bar{l}_i)
\end{aligned}$$

Table 1(a) Lower blocks of symmetric adders with formats 1 to 7

$$\begin{aligned}
8. \quad m_i &= \kappa_i \bar{k}_i \oplus \lambda_i \bar{l}_i \\
d_{i-1} &= m_{i-1} \oplus (\bar{\kappa}_i \bar{\lambda}_i \vee \bar{k}_i \bar{\lambda}_i \vee \bar{\kappa}_i \bar{l}_i \\
&\quad \vee \kappa_i k_i \lambda_i l_i) \\
b_{i-2} &= \kappa_i k_i \lambda_i l_i \vee m_{i-1} (\kappa_i k_i \vee \kappa_i \lambda_i \\
&\quad \vee \bar{k}_i \lambda_i \bar{l}_i)
\end{aligned}$$

$$\begin{aligned}
9. \quad m_i &= \kappa_i \bar{k}_i \oplus \lambda_i \bar{l}_i \\
d_{i-1} &= m_{i-1} \oplus (\kappa_i \oplus k_i \oplus \lambda_i \oplus \bar{l}_i) \\
b_{i-2} &= \lambda_i l_i (\kappa_i \vee \bar{k}_i) \vee \bar{\lambda}_i \bar{l}_i (\kappa_i \oplus k_i) \\
&\quad \vee m_{i-1} ((\kappa_i \oplus k_i) \vee (\lambda_i \oplus \bar{l}_i))
\end{aligned}$$

Table 1(b) Lower blocks of symmetric adders with formats 8 and 9

$$\begin{aligned}
1. \quad m_i &= \kappa_i \oplus k_i \oplus \lambda_i \oplus l_i \\
d_i &= m_{i-1} \oplus (\kappa_i \bar{k}_i \bar{l}_i \vee \kappa_i \bar{k}_i \lambda_i \vee \bar{k}_i \lambda_i \bar{l}_i \\
&\quad \vee \kappa_i \lambda_i \bar{l}_i \vee \bar{\kappa}_i k_i \bar{\lambda}_i l_i) \\
b_{i-2} &= \bar{\kappa}_i k_i \bar{\lambda}_i l_i \vee m_{i-1} (\bar{\lambda}_i l_i \vee \bar{\kappa}_i k_i \vee \bar{k}_i \bar{\lambda}_i) \\
&\quad \vee \bar{\kappa}_i \bar{\lambda}_i \vee \bar{\kappa}_i l_i)
\end{aligned}$$

$$\begin{aligned}
2. \quad m_i &= k_i \oplus l_i \\
d_{i-1} &= m_{i-1} \oplus (\kappa_i \lambda_i \vee \bar{k}_i \bar{l}_i \vee \bar{k}_i \lambda_i \\
&\quad \vee \bar{\kappa}_i k_i \bar{\lambda}_i l_i) \\
b_{i-2} &= \bar{\kappa}_i k_i \bar{\lambda}_i l_i \vee m_{i-1} (\bar{\lambda}_i l_i \vee \bar{\kappa}_i k_i \vee \bar{\kappa}_i \bar{\lambda}_i)
\end{aligned}$$

$$\begin{aligned}
3. \quad m_i &= k_i \oplus l_i \\
d_{i-1} &= m_{i-1} \oplus (\kappa_i \lambda_i l_i \vee \bar{k}_i \lambda_i l_i \vee \kappa_i k_i \lambda_i \\
&\quad \vee \bar{\kappa}_i k_i \bar{\lambda}_i l_i) \\
b_{i-2} &= \bar{\kappa}_i k_i \bar{\lambda}_i l_i \vee m_{i-1} (\kappa_i k_i \vee \bar{\lambda}_i l_i \vee k_i l_i) \\
&\quad \vee m_{i-1} (\bar{\kappa}_i k_i \vee \bar{\lambda}_i l_i \vee k_i l_i)
\end{aligned}$$

$$\begin{aligned}
4. \quad m_i &= \bar{\kappa}_i \bar{k}_i \oplus \bar{\lambda}_i \bar{l}_i \\
d_{i-1} &= m_{i-1} \oplus (k_i l_i \vee \kappa_i \bar{l}_i \vee \bar{k}_i \lambda_i) \\
b_{i-2} &= k_i l_i \vee m_{i-1} (k_i \vee l_i \vee \bar{\kappa}_i \bar{\lambda}_i)
\end{aligned}$$

$$\begin{aligned}
5. \quad m_i &= k_i \oplus l_i \\
d_{i-1} &= m_{i-1} \oplus ((\kappa_i \oplus \lambda_i) \vee \bar{k}_i \bar{l}_i) \\
b_{i-2} &= \bar{\kappa}_i \bar{\lambda}_i \vee m_{i-1} (\bar{\kappa}_i \vee \bar{\lambda}_i \vee \bar{k}_i \bar{l}_i)
\end{aligned}$$

Table 2(a) Lower blocks of symmetric subtractors with formats 1 to 5

$$\begin{aligned}
6. \quad m_i &= \bar{\kappa}_i \bar{k}_i \oplus \bar{\lambda}_i \bar{l}_i \\
d_{i-1} &= m_{i-1} \oplus ((\kappa_i \oplus k_i) (\lambda_i \oplus l_i) \\
&\quad \vee \kappa_i k_i (\bar{\lambda}_i \oplus \bar{l}_i) \vee \bar{\kappa}_i \bar{k}_i \bar{\lambda}_i \bar{l}_i) \\
b_{i-2} &= (\kappa_i \oplus k_i) (\lambda_i \oplus l_i) \\
&\quad \vee m_{i-1} ((\lambda_i \oplus l_i) \vee (\kappa_i \oplus k_i) \vee \kappa_i \lambda_i)
\end{aligned}$$

$$\begin{aligned}
7. \quad m_i &= \bar{\kappa}_i \bar{k}_i \oplus \bar{\lambda}_i \bar{l}_i \\
d_{i-1} &= m_{i-1} \oplus (\kappa_i \lambda_i \vee \bar{k}_i \lambda_i \vee \kappa_i \bar{l}_i \\
&\quad \vee \bar{\kappa}_i k_i \bar{\lambda}_i l_i) \\
b_{i-2} &= \bar{\kappa}_i k_i \bar{\lambda}_i l_i \vee m_{i-1} (\bar{\kappa}_i k_i \vee \bar{\lambda}_i l_i \vee \bar{\kappa}_i \bar{\lambda}_i)
\end{aligned}$$

$$\begin{aligned}
8. \quad m_i &= \kappa_i \bar{k}_i \oplus \lambda_i \bar{l}_i \\
d_{i-1} &= m_{i-1} \oplus (\bar{\kappa}_i \bar{\lambda}_i \vee m_{i-1} ((\kappa_i \oplus \lambda_i) \\
&\quad \vee \kappa_i k_i \bar{\lambda}_i l_i) \\
b_{i-2} &= \bar{\kappa}_i \bar{\lambda}_i \vee m_{i-1} (\bar{\kappa}_i \vee \bar{\lambda}_i \vee \kappa_i \bar{k}_i \lambda_i \bar{l}_i)
\end{aligned}$$

$$\begin{aligned}
9. \quad m_i &= \kappa_i \bar{k}_i \oplus \lambda_i \bar{l}_i \\
d_{i-1} &= m_{i-1} \oplus (\bar{\kappa}_i k_i \bar{\lambda}_i l_i \vee (\lambda_i \vee \bar{l}_i) \\
&\quad \cdot (\kappa_i \oplus k_i) \vee \kappa_i (\bar{\lambda}_i \oplus \bar{l}_i)) \\
b_{i-2} &= \bar{\kappa}_i k_i \bar{\lambda}_i l_i \vee m_{i-1} (\bar{\kappa}_i k_i \vee \bar{\lambda}_i l_i \\
&\quad \vee \kappa_i \bar{k}_i \lambda_i \bar{l}_i)
\end{aligned}$$

Table 2(b) Lower blocks of symmetric subtractors with formats 6 to 9

### References

1. L.N. Goyal, "A study in the design of an arithmetic element for serial processing in a linear iterative structure," Report 797, Dept. of Computer Sc., Univ. of Illinois, Urbana, 1976.
2. M.D. Ercegovic, "A general method for evaluation of functions and computations in a digital computer," Report 750, Dept. of Computer Sc., Univ. of Illinois, Urbana, 1975.
3. A. Avizienis, "Signed-digit number representations for fast parallel arithmetic," IRE Transaction on Electronic Computers, 1961, p.389.
4. R.T. Borovec, "The logical design of a class of limited carry-borrow propagation adders," Report 275, Dept. of Computer Sc., Univ. of Illinois, Urbana, 1968.
5. J.E. Robertson, "A deterministic procedure for the design of carry-save adders and borrow-save subtractors," Report 235, Dept. of Computer Sc., Univ. of Illinois, Urbana, 1967.
6. F.A. Rohatsch, "A study of transformations applicable to the development of limited carry-borrow propagation adders," Report 226, Dept. of Computer Sc., Univ. of Illinois, Urbana, 1967.